# UM MODELO DE SISTEMA TUTOR INTELIGENTE APLICADO AO ENSINO DA PROGRAMAÇÃO ESTRUTURADA

UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

# UM MODELO DE SISTEMA TUTOR INTELIGENTE APLICADO AO ENSINO DA PROGRAMAÇÃO ESTRUTURADA

Élcio Miguel Prus

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia da Produção da Universidade Federal de Santa Catarina como requisito parcial para a obtenção do título de Mestre em Engenharia de Produção.

Florianópolis
2001

Élcio Miguel Prus

**UM MODELO DE UM SISTEMA TUTOR INTELIGENTE APLICADO AO ENSINO DA PROGRAMAÇÃO ESTRUTURADA**

Esta dissertação foi julgada e aprovada para obtenção
do título de **Mestre em Engenharia de Produção** no
**Programa de Pós-Graduação em Engenharia de Produção**
da Universidade Federal de Santa Catarina

Florianópolis, 28 de maio de 2001.

Prof. Ricardo Miranda Barcia, PhD
Coordenador do Curso

BANCA EXAMINADORA

Profª. Alice Theresinha Cybis Pereira, PhD
Orientadora

Profª. Marta Costa Rosatelli, Dra.

Prof. Fernando Álvaro Ostuni Gauthier , Dr.

A minha esposa Carem e aos filhos Ivens e Igor pela compreensão e sacrifício que, carinhosamente, cederam e que me guiaram nos momentos mais difíceis desta caminhada.

## AGRADECIMENTOS

A Deus pela criação e proteção.

Aos meus pais pela existência e perseverança em me guiar para o melhor caminho.

A minha querida esposa e filhos pelo amor e compreensão dedicados durante a realização desta dissertação.

Aos familiares, amigos e companheiros pela colaboração, muitas vezes decisiva.

A Professora PhD Alice Theresinha Cybis Pereira pelo valioso apoio dado para que esta dissertação se concretizasse.

A todos aqueles que não foram citados e que contribuíram, direta ou indiretamente, para a realização desta pesquisa.

" Uma pessoa realmente grande é aquela que nos dá uma chance. "

Paul Duffy

## SUMÁRIO

LISTA DE FIGURAS ..... ix
LISTA DE QUADROS ..... x
LISTA DE TABELAS ..... xi
LISTA DE REDUÇÕES ..... xii
RESUMO ..... xiii
ABSTRACT ..... xv

1 INTRODUÇÃO ..... 1
1.1 Identificação do Problema ..... 3
1.2 Questão da Pesquisa ..... 5
1.3 Justificativa ..... 5
1.4 Objetivos ..... 8
1.4.1 Objetivo Geral ..... 8
1.4.2 Objetivos Específicos ..... 8
1.5 Metodologia ..... 9
1.6 Limitações da Pesquisa ..... 9
1.7 Estrutura da Dissertação ..... 10

2 ALGORITMOS ESTRUTURADOS ..... 12
2.1 Considerações Iniciais ..... 12
2.2 Definição de Algoritmo ..... 13
2.3 Estrutura Básica de um Algoritmo ..... 16
2.3.1 Variáveis ..... 17
2.4 Estruturas ..... 18
2.4.1 Estrutura SE-ENTÃO-SENÃO ..... 19
2.4.2 Estrutura ESCOLHA CASO ..... 20
2.4.3 Estrutura ENQUANTO-FAÇA ..... 21
2.4.4 Estrutura PARA-FAÇA ..... 22
2.4.5 Estrutura REPITA-ATÉ ..... 23
2.5 Considerações Finais ..... 24

3 INTELIGÊNCIA ARTIFICIAL ..... 26
3.1 Inteligência Artificial ..... 26
3.2 Inteligência Artificial aplicada na Educação ..... 30
3.3 Sistemas Tutores Inteligentes ..... 32
3.3.1 Estrutura de um STI ..... 34
3.3.1.1 Modelo do Especialista ..... 35
3.3.1.2 Modelo do Estudante ..... 36
3.3.1.3 Modelo Pedagógico ..... 37
3.3.1.4 Modelo de Interface ..... 38
3.4 Formas de Representação do Conhecimento ..... 39
3.4.1 Procedimentos ..... 41
3.4.2 Esquemas ..... 42

3.4.2.1 Scripts ..... 42
3.4.2.2 Frame ..... 46
3.4.3 Redes Semânticas ..... 48
3.5 Exemplos de STI aplicados a Programação Estruturada ..... 49
3.6 Exemplo de STI relacionado ao Modelo Proposto ..... 51
3.7 Considerações Finais ..... 52

4 TEORIAS DE APRENDIZAGEM ..... 55
4.1 Considerações Iniciais ..... 55
4.2 Empirismo ..... 57
4.2.1 Aprendizagem ..... 60
4.2.2 Relacionamento Professor-Aluno ..... 62
4.2.3 Metodologia ..... 63
4.2.3.1 Ensino individualizado ..... 64
4.2.3.2 Ensino baseado na competência ..... 64
4.2.3.3 Ensino programado ..... 65
4.2.4 Avaliação ..... 66
4.3 Inatismo ..... 66
4.3.1 Aprendizagem ..... 70
4.3.2 Relacionamento Professor-Aluno ..... 71
4.3.3 Metodologia ..... 73
4.3.4 Avaliação ..... 73
4.4 Interacionismo ..... 74
4.4.1 Construtivismo ..... 75
4.4.1.1 Aprendizagem ..... 79
4.4.1.2 Relacionamento Professor-Aluno ..... 80
4.4.1.3 Metodologia ..... 81
4.4.1.4 Avaliação ..... 82
4.4.2 Sócio-Interacionismo ..... 83
4.4.2.1 Aprendizagem ..... 87
4.4.2.2 Relacionamento Professor-Aluno ..... 88
4.4.2.3 Metodologia ..... 89
4.4.2.4 Avaliação ..... 89
4.5 Considerações Finais ..... 90

5 MODELO PROPOSTO ..... 92
5.1 Fundamentos do Modelo Proposto ..... 92
5.2 Caracterização do Modelo ..... 95
5.3 Estrutura e Funcionamento do Modelo ..... 97
5.3.1 Professor ..... 98
5.3.2 O Modelo do Especialista ..... 101
5.3.3 O Modelo Pedagógico ..... 105
5.3.4 O Modelo de Interface ..... 106
5.3.5 O Modelo do Estudante ..... 112
5.4 Considerações Finais ..... 116

6 CONCLUSÕES E RECOMENDAÇÕES ..... 118
6.1 Avaliação do Modelo Proposto ..... 118

6.1.1 Aspectos Positivos ..... 118
6.1.2 Aspectos Negativos ..... 120
6.2 Conclusões ..... 121
6.3 Recomendações ..... 122

7 REFERÊNCIAS BIBLIOGRÁFICAS ..... 124

8 ANEXOS ..... 129
8.1 Anexo 1 – Conteúdo Programático da Algoritmos e Programação I ..... 129

# LISTA DE FIGURAS

Figura 1: Diagrama em blocos de um STI básico ..... 34
Figura 2: Exemplo de Rede Semântica ..... 48
Figura 3: Exemplo de Rede Semântica ..... 49
Figura 4: Diagrama em blocos do Modelo Proposto de STI ..... 97
Figura 5: Tela do Professor ..... 98
Figura 6: Opções do Menu do Módulo do Professor ..... 99
Figura 7: Tela de Cadastro de Professor ..... 99
Figura 8: Tela de Cadastro de Turma ..... 100
Figura 9: Tela de Cadastro de Disciplina ..... 101
Figura 10: Relacionamento entre teoria e exercícios ..... 102
Figura 11: Tela de Cadastro de Teoria ..... 103
Figura 12: Tela de Cadastro de Exercício ..... 104
Figura 13: Tela do Aluno ..... 106
Figura 14: Tela de Cadastro do Aluno ..... 107
Figura 15: Tela de Resolução dos Exercícios ..... 108
Figura 16: Tela de Teoria e Solução do Professor ..... 111
Figura 17: Representação gráfica da situação do aluno A ..... 113
Figura 18: Representação gráfica da situação do aluno B ..... 114
Figura 19: Representação gráfica da situação do aluno C ..... 115
Figura 20: Representação gráfica da situação do aluno D ..... 116

# LISTA DE QUADROS

**Quadro 1: Tipos de Dados..... 17**

# LISTA DE TABELAS

**Tabela 1: Regras para escolha do próximo exercício..... 110**

# LISTA DE REDUÇÕES

**Siglas**

| | |
|---|---|
| CAI | *Computer Assisted Instruction* |
| DER | Diagrama Entidade Relacionamento |
| ENC | Exame Nacional de Cursos |
| IA | Inteligência Artificial |
| IAED | Inteligência Artificial aplicada a Educação |
| ICAI | *Inteligent Computer Assisted Instruction* |
| IndAp | Índice de Aproveitamento |
| LDB | Lei de Diretrizes e Bases da Educação Nacional |
| MEC | Ministério da Educação e Cultura |
| OOP | Programação Orientada à Objetos |
| RBC | Raciocínio Baseado em Casos |
| SBC | Sociedade Brasileira de Computação |
| STI | Sistemas Tutores Inteligentes |
| TPD | Tecnologia em Processamento de Dados |
| WEI | Workshop de Educação em Informática |
| ZDP | Zona de Desenvolvimento Proximal |

# RESUMO

PRUS, Élcio Miguel. Um modelo de sistema tutor inteligente aplicado ao ensino da programação estruturada. Florianópolis, 2001. 130 f. Dissertação (Mestrado em Engenharia de Produção) – Programa de Pós-graduação em Engenharia de Produção. UFSC, 2001.

Este trabalho apresenta um modelo e um protótipo, desenvolvido em Delphi® 5.0, de sistema tutor inteligente aplicado ao ensino de algoritmos estruturados, bem como a sua avaliação, utilizando a disciplina de Algoritmos e Programação I, no curso de Tecnologia em Processamento de dados, ofertado pelas Faculdades Santa Cruz, em Curitiba – Paraná.

A pesquisa foi baseada em revisão bibliográfica e no desenvolvimento de um modelo e implementação de um protótipo. São apresentados fundamentos teóricos sobre algoritmos estruturados, inteligência artificial em geral e aplicada à educação, assim como sistemas tutores inteligentes, com alguns exemplos relacionados com a linha de pesquisa da dissertação. É feita uma análise das principais correntes de teorias de aprendizagem, visando a fundamentar o desenvolvimento do protótipo.

A estrutura do modelo proposto e do protótipo apresentado é dividido em dois módulos: o do professor e do aluno. A utilização de exemplos é apresentada para melhor ilustrar a aplicação do modelo ao contexto de ensino de algoritmo estruturado.

Como conclusão, a construção do protótipo e sua aplicação em parte da turma é o resultado concreto deste trabalho e suas partes ressaltam a importância do modelo desenvolvido.

**Palavras-chave:** Algoritmos Estruturados, Inteligência Artificial, Sistemas Tutores Inteligentes, Teorias de Aprendizagem.

# ABSTRACT

PRUS, Élcio Miguel. Um modelo de sistema tutor inteligente aplicado ao ensino da programação estruturada. Florianópolis, 2001. 130 f. Dissertação (Mestrado em Engenharia de Produção) – Programa de Pós-graduação em Engenharia de Produção. UFSC, 2001.

A model and a prototype are introduced by this dissertation. Delphi® 5.0 programming language is used to compound the prototype. Santa Cruz College, in Curitiba, Paraná, has a Data Processing Technology Course and this forthcoming presented model is an intelligent tutor system applied to teaching and evaluating process of structure algorithms, practiced in Algorithms and Programming I discipline.

The research was based on a bibliographical review and on the model development and the prototype implementation, theoretical support about structured algorithm, generical aspects and educational applications of artificial intelligence, as well as intelligent tutor systems. Some examples are made part of this dissertation research field.

An analysis is proceeded over the main theoretical learning approaches, aiming to make the prototype foundation development possible.

The frame of this purpose, considering the model and the prototype, is divided into two parts: the teacher and the student. The examples are used to let the contextual learning participate consistently.

The goal is achieved with the prototype construction and its application in part of a student's group. It confirms the relevance of the model in use.

**Key-words: Structured Algorithms, Artificial Intelligence, Intelligent Tutoring Systems, Theories of Learning.**

# 1 INTRODUÇÃO

A qualidade do ensino oferecido pelas instituições de nível superior no Brasil é a constante preocupação do Ministério da Educação e Cultura (MEC). Esta excelência de ensino, meta intrínseca ao artigo 208, inciso V, da Constituição Federal e premissa básica da nova Lei de Diretrizes e Bases da Educação Nacional (LDB), em vigor desde dezembro de 1996, só poderá ser alcançada através de um ensino moderno, atualizado e que vise a formação profissional (Niskier, 1996).

Esta preocupação com a qualidade do ensino oferecido pelas instituições de nível superior não é em vão. O Exame Nacional de Cursos (ENC), conhecido como provão, tem por objetivo avaliar, logo após a conclusão do curso, os conhecimentos adquiridos pelos graduados durante o desenvolvimento do mesmo. Este revelou em sua última edição, 1999, que nove instituições das trinta e seis que oferecem o curso de administração, no estado do Paraná, ou seja 25%, tiveram conceito D e E, considerados os dois mais baixos (Ministério da Educação e Cultura, 1999).

Com base nesta nova realidade o MEC, através da Sociedade Brasileira de Computação (SBC), promoveu em junho de 1998, em Belo Horizonte, Minas Gerais, um Workshop de Educação em Computação (WEI/98). Neste foram discutidas e oficializadas as diretrizes curriculares referentes aos cursos, de

nível superior, de informática e computação. Estas diretrizes fornecem às instituições parâmetros que norteiam a formulação dos currículos.

Estas diretrizes também normalizam as denominações dos cursos superiores em informática, evitando ambigüidade, facilitando a comparação entre os diversos cursos existentes no país e principalmente, dando um passo definitivo a criação dos conselhos federal e regional que regulamentarão as atividades referentes a profissão, ainda não existentes.

Segundo estas diretrizes, os currículos dos cursos da área de computação e informática são compostos por quatro grandes áreas de formação:

- formação básica que compreende os princípios básicos da área de computação, a ciência da computação, a matemática necessária para definí-los formalmente, a física e eletricidade necessária para permitir o entendimento e o projeto de computadores viáveis tecnicamente e a formação pedagógica que introduz os conhecimentos básicos da construção do conhecimento, necessários ao desenvolvimento da prática do ensino de computação;
- formação tecnológica (também chamada de aplicada ou profissional) que aplica os conhecimentos básicos no desenvolvimento tecnológico da computação;
- formação complementar que permite uma interação dos egressos dos cursos com outras profissões;

- formação humanística que dá ao egresso uma dimensão social e humana.

Esta divisão proporciona uma maior flexibilidade na elaboração das grades curriculares, e facilita a modificação do enfoque de algumas disciplinas, de acordo com a evolução tecnológica. A formação básica é a principal fundamentação que o novo profissional deve possuir. Ela é composta por disciplinas da área de programação, computação e algoritmos.

A programação, entendida como programação de computadores, é uma atividade voltada à solução de problemas. Nesse sentido ela está relacionada com uma variada gama de outras atividades como especificação, projeto, validação, modelagem e estruturação de programas e dados, utilizando-se das linguagens de programação propriamente ditas, como ferramentas.

Ao contrário do que se apregoava há alguns anos atrás, a atividade de programação deixou de ser uma arte para se tornar uma ciência, envolvendo um conjunto de princípios, técnicas e formalismos que visam a produção de *software* bem estruturado e confiável (Salvetti & Barbosa, 1998).

### 1.1 Identificação do Problema

Uma das disciplinas que pertencem ao currículo de formação básica é a de Algoritmo e Programação. Com ela o egresso, ou recém-formado, será capaz

de produzir programas, de média complexidade, para a resolução de problemas que utilizem o computador e seus recursos como meio para tal.

O aluno ao concluir esta disciplina deve possuir domínio de todas as técnicas para o planejamento, desenvolvimento e manutenção de programas desenvolvidos com a técnica de programação estruturada. É importante também possuir conhecimentos da metodologia da engenharia de *software*, para que sua visão seja mais ampla e não se restrinja a resolução, apenas, de um único problema.

Seguindo esta linha de raciocínio, é ministrado, nas Faculdades Regional Santa Cruz, em Curitiba, Paraná, dentro do Curso de Tecnologia em Processamento de Dados (TPD), no primeiro semestre, a disciplina de Algoritmos e Programação I. Esta tem como objetivo desenvolver as noções básicas de programação, com base nas técnicas e estruturas da programação estruturada, aplicando-as em algoritmos para a resolução de problemas matemáticos, através dos conceitos do português estruturado.

Esta disciplina é considerada uma disciplina introdutória a programação e pré-requisito para todas as outras que envolvem técnicas de programação. Por trabalhar com conceitos não tão comuns ao cotidiano do aluno, esta disciplina encontra diversas barreiras em sua compreensão. Segundo Direne & Pimentel (1998) do ponto de vista de fatores cognitivos, a programação de

computadores é uma tarefa de alta carga para iniciantes por duas razões principais: a falta do conhecimento de princípios de programação e a falta de perícia.

### 1.2 Questão da Pesquisa

" Como facilitar ao estudante iniciante de computação a apreensão dos conceitos de algoritmo e programação ? "

### 1.3 Justificativa

Diante do problema da dificuldade dos discentes em desenvolver algoritmos, e em conseqüência, transformá-los em programas, torna-se necessário alguma ferramenta auxiliar às aulas para facilitar a aquisição do conhecimento por parte dos alunos e garantir que os mesmos dominem as técnicas utilizadas na construção de algoritmos (Viccari, 1996).

Segunda Giraffa (1998) normalmente em uma aula de programação de computadores, ministrada em sala de aula ou em um laboratório de informática, após terminar a explicação do assunto e a exemplificação utilizada, o professor escolhe um exercício para seus alunos desenvolverem. A questão está em como e qual exercício selecionar. Usando da sua percepção e das variáveis que o professor utiliza para avaliar a turma, como compreensão do assunto, notas na última avaliação, participação em aulas, entre outras, ele escolhe um exercício que atenda e acompanhe o momento demonstrado pela turma. Para

que esta escolha esteja próxima de uma turma real, em um sistema tutor inteligente (STI), a escolha deve seguir critérios que avaliem a condição atual de conhecimento do aluno, ou seja, o STI deve possuir um mecanismo que proporcione medidas cognitivas para escolha dos próximos exercícios.

Medidas Cognitivas para a escolha adequada de um próximo exemplo foram apresentadas em Pimentel & Direne (1997) para o ensino de conceitos visuais aplicados à radiologia médica. Tanto o ensino de conceitos visuais quanto o ensino de programação de computadores são baseados na aquisição de perícia e na apresentação de casos para os alunos solucionarem. Porém, em radiologia médica a solução é um diagnóstico expresso em um laudo, enquanto que na programação a solução é um algoritmo expresso em um programa de computador.

Para Direne & Pimentel (1998) a utilização de STI destinados a iniciantes de programação tentam lidar com estas duas carências por meio da combinação de:

- técnicas de Inteligência Artificial;
- ambientação construtivista por exploração livre;
- recursos de visualização científica para, no final, criar interfaces de razoável versatilidade para o aluno.

Em pesquisa verbal realizada em 20 de setembro de 2000 com sessenta e oito

alunos da disciplinas de Algoritmos e Programação I, sessenta e dois, ou seja 91%, declararam possuir e utilizar freqüentemente computador em suas residências ou no trabalho. O restante de seis alunos (9%) declararam que pretendem adquirir um equipamento nos próximos doze meses.

Com este levantamento fica evidente o perfil dos alunos que freqüentam o curso. Todos eles sabem da importância e da necessidade do computador no desenvolvimento de sua profissão. Em síntese, o computador, já é realidade para 91 % dos alunos pesquisados.

Ainda dentro da grade do primeiro semestre do curso de TPD é ministrada a disciplina de Laboratório I que tem como objetivo familiarizar o aluno ao uso dos recursos mínimos do computador, tanto em programas aplicativos como aos periféricos disponíveis. Esta disciplina juntamente com o resultado da pesquisa fortalecem a premissa que o aluno do primeiro semestre, em sua grande maioria, já possui conhecimento do computador como ferramenta, e os que ainda não o fazem, terão a oportunidade de aprender durante o primeiro semestre do curso.

Diante destas colocações tem-se um ambiente favorável para o desenvolvimento de um modelo de STI que facilitará a aprendizagem da programação estruturada e que será de grande auxílio no desenvolvimento da disciplina de Algoritmos e Programação I.

## 1.4 Objetivos

Para o desenvolvimento deste trabalho de pesquisa foi delineado o seguinte objetivo geral e os seus desmembramentos em objetivos específicos.

### 1.4.1 Objetivo Geral

Projetar e desenvolver um modelo de STI que auxilie no aprendizado das estruturas básicas de programação, utilizando os conceitos de algoritmos, exemplos e proporcionando séries de exercícios apresentadas através de diversas estratégias de aprendizagem e de acordo com o aprendizado demonstrado pelo aluno.

### 1.4.2 Objetivos Específicos

Desenvolver o modelo de um ambiente inteligente, interativo e dinâmico que:

- Proporcione automação na escolha de um próximo algoritmo a ser desenvolvido pelo aluno, observando as condições atual de conhecimento;
- Apresente diversas visões do mesmo conceito, tornando a base teórica da disciplina ampla e interdisciplinar;
- Apresente conceitos, com teoria e exemplos, através de diversas teorias de aprendizagem;
- Disponibilize uma revisão conceitual, caso o aluno durante o desenvolvimento de algum algoritmo apresente dificuldades de elaboração;
- Proporcione uma série de exemplos de cada estrutura de programação, classificada por grau de dificuldade, onde à medida que o aluno resolva um

algoritmo, o próximo sugerido será de acordo com o grau de resolução (dificuldade) apresentado no anterior;

- Seja aplicado à disciplina de Algoritmos e Programação I, no curso de Tecnologia em Processamento de Dados (TPD), das Faculdades Santa Cruz de Curitiba.

## 1.5 Metodologia

Esta pesquisa é de natureza aplicada pois objetiva gerar um modelo de STI para contribuir na solução do problema de aprendizagem de algoritmo e programação. Está baseada em:

- Pesquisa bibliográfica;
- Idealização e desenvolvimento de um modelo.

## 1.6 Limitações da Pesquisa

Esta dissertação está concentrada na disciplina de Algoritmos e Programação I, ministrada no 1° semestre do curso de Tecnologia em Processamento de Dados das Faculdades Santa Cruz de Curitiba. Outras instituições de ensino superior de informática ministram esta disciplina, mas não se pode precisar se com o mesmo conteúdo. Portanto todo o direcionamento da pesquisa está voltado para o curso de Tecnologia em Processamento de Dados das Faculdades Santa Cruz de Curitiba.

A aplicação desenvolvida com base no modelo proposto está direcionada à

disciplina e instituição acima mencionadas. Não foi objetivo desta pesquisa a análise de currículos de algoritmos e programação ministrados por outras instituições de ensino.

### 1.7 Estrutura da Dissertação

Esta dissertação está estruturada em sete capítulos. No primeiro encontram-se a introdução, a identificação do problema, a questão da pesquisa, a justificativa do trabalho, os objetivos gerais e específicos, a metodologia aplicada e as limitações do trabalho.

No segundo capítulo inicia-se a revisão bibliográfica com direcionamento explicativo sobre Algoritmos Estruturados. O principal objetivo deste capítulo é demonstrar os principais conceitos e definições, considerando os principais autores da área. Será descrito neste capítulo a estrutura utilizada em algoritmos e as estruturas básicas de programação. Os exemplos são demonstrados na linguagem conhecida como pseudocódigo.

No capítulo 3 encontra-se a revisão bibliográfica sobre Inteligência Artificial (IA), Inteligência Artificial aplicada a Educação (IAED), STI e Formas de Representação do Conhecimento. Neste capítulo encontra-se as principais definições, conceitos e exemplos aplicados a cada um dos temas acima relacionados.

No capítulo 4 temos a revisão bibliográfica sobre as principais teorias de aprendizagem, como estão estruturadas em relação aprendizagem, relacionamento professor-aluno, metodologia aplicada e forma de avaliação. O principal objetivo deste capítulo é descrever as principais correntes do pensamento para os modelos educacionais, com base nas idéias das obras de seus representantes mais relevantes, fundamentando as teorias de aprendizagem para serem utilizadas no módulo pedagógico do modelo do STI proposto.

O quinto capítulo refere-se ao desenvolvimento do modelo proposto, estando estruturado primeiramente na sua fundamentação e caracterização. Neste capítulo encontra-se as principais telas do protótipo desenvolvido a partir do modelo proposto com explicações que exemplificam algumas situações possíveis de acontecer na utilização do protótipo.

No sexto capítulo estão a avaliação do protótipo desenvolvido, as conclusões e recomendações da dissertação, e finalizando no capítulo sete estão as fontes bibliográficas utilizadas nesta dissertação. Estas referências estão relacionadas aos conteúdos de algoritmos estruturados e lógica de programação, aos conteúdos de IA, IAED e STI e Formas de Representação do Conhecimento e aos conteúdos de teorias de aprendizagem.

# 2 ALGORITMOS ESTRUTURADOS

O objetivo deste capítulo é demonstrar os principais conceitos e definições sobre algoritmos estruturados. O capítulo inicia-se com definições de algoritmos compiladas de autores da área. Na seqüência está descrita a estrutura utilizada na construção de um algoritmo e as estruturas básicas de programação, divididas em estruturas de controle e de decisão. Os exemplos apresentados estão demonstrados na linguagem conhecida como pseudocódigo.

## 2.1 Considerações Iniciais

A automatização de tarefas é um aspecto marcante da sociedade moderna. O aperfeiçoamento tecnológico alcançado teve, como elementos fundamentais, a análise e a obtenção de descrições da execução de tarefas em termos de ações simples o suficiente, tal que pudessem ser automatizadas por uma máquina especialmente desenvolvida para este fim, o computador (Cormen, 1990).

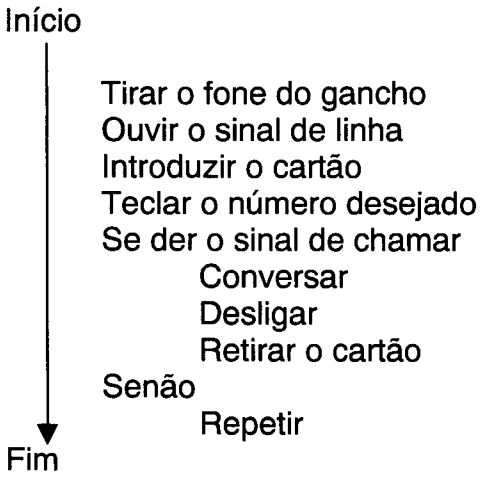
Para que esta automatização ocorra é necessário uma seqüência de instruções que fará com que o computador realize determinada tarefa. Segundo Tremblay & Bunt (1983) esta seqüência de instruções não deve possibilitar a interpretação alternativa que possa fazer com que o computador tome um caminho diferente daquele inicialmente planejado.

Este cuidado na formulação das instruções e na sua estruturação é o alvo de estudo dos algoritmos, parte da ciência da computação que desenvolve e aprimora técnicas de construção de programas de forma a determinar que o computador siga pelo único caminho correto possível que conduza aos resultados desejados.

## 2.2 Definição de Algoritmo

O algoritmo pode ser usado como uma ferramenta genérica para representar a solução de tarefas independente do desejo de automatizá-las, mas em geral está associado ao processamento eletrônico de dados, onde representa o rascunho para programas ou *softwares*.

Apesar do termo ser novo em si, o conceito é certamente bastante familiar. As indicações dadas para se chegar até uma determinada rua constituem um algoritmo. Uma receita de cozinha é uma forma muito familiar de algoritmo. Uma planta de obra serve ao mesmo propósito num projeto de construção. Estes algoritmos são conhecidos como não computacionais. Um exemplo de um algoritmo não computacional cujo objetivo é usar um telefone público é apresentado a seguir:




Salveti (1998) define um algoritmo como uma seqüência finita de instruções ou operações básicas (operações definidas sem ambigüidade e executáveis em tempo finito) cuja finalidade é resolver um problema computacional.

Segundo Pinto (1990) um algoritmo é uma receita para um processo computacional e consiste de uma série de operações primitivas, interconectadas devidamente, sobre um conjunto de objetos. Os objetos manipulados por essas receitas são as variáveis.

Para Terada (1991) algoritmo é uma descrição passo a passo de como um problema é solucionável. Esta descrição deve ser finita, e os passos devem ser bem definidos e sem ambigüidades. Warnier (1991) ressalta que a característica mais importante de um algoritmo é a simplicidade e a isenção de ambigüidade. As instruções devem estar numa ordem cuidadosamente definida e o algoritmo deve ser efetivo, isto é, deve sempre resolver um problema utilizando um número finito de instruções.

Szwarcfiter & Markezon (1994) define algoritmo como um processo sistemático para a resolução de um problema. O desenvolvimento de algoritmos é particularmente importante para problemas a serem solucionados em um computador, pela própria natureza do instrumento utilizado. Dois aspectos devem ser considerados no desenvolvimento de algoritmos: a correção e a análise.

O primeiro consiste em verificar a exatidão do método empregado, o que é realizado através de uma prova matemática. A análise visa a obtenção de parâmetros que possam avaliar a eficiência do algoritmo em termos de tempo de execução e memória ocupada. A análise é realizada através de um estudo do comportamento do algoritmo.

Como qualquer modelo, um algoritmo é uma abstração da realidade. Wirth (1989) define abstração como uma simplificação dos fatos constantes da realidade que se encontra o problema a ser resolvido. Os dados utilizados nos algoritmos representam algumas propriedades e características dos fatos reais, pois outras são desprezadas, por serem inexpressivas ou irrelevantes para a solução adotada.

Todas as definições apresentadas de algoritmo nos conduzem a descrição de soluções de problemas do mundo real, para serem implementadas utilizando os recursos computacionais.

Como o mundo computacional possui severas limitações em relação ao real, exige-se que, sejam impostas algumas regras e utilizadas algumas estruturas adequadas para a solução de problemas. Estas estruturas possibilitam a simulação de rotinas, de decisões e de formas de controle.

## 2.3 Estrutura Básica de um Algoritmo

Para que o computador possa compreender os passos necessários para a resolução de um determinado problema, devemos estruturar estes passos de forma que sejam perfeitamente compreendidos e executados.

Esta organização das ações a serem tomadas pela máquina de forma organizada e lógica possui uma rígida regra de sintaxe e semântica. Para Pinto (1990) sintaxe é um conjunto de regras formais, que especificam a composição de algoritmos a partir de letras, dígitos e outros símbolos. As regras de semântica especificam o significado de qualquer algoritmo sintaticamente válido, escrito em uma linguagem.

Segundo Salvetti & Barbosa (1998), Pinto (1990), Farrer (1999) e Guimarães (1985) a linguagem mais adequada para representar um algoritmo é a conhecida como pseudocódigo.

Em pseudocódigo a estrutura de um algoritmo é:

```
ALGORITMO <<nome do algoritmo>>
        <<definições das variáveis>>
INÍCIO
        <<comandos 1>>
        <<comandos 2>>
        <<comandos N>>

FIM
```

Esta sintaxe permite definir as variáveis que serão utilizadas na resolução do problema e os comandos que serão aplicados para que possam processar as informações de entrada transformando-as em informações de saída.

### 2.3.1 Variáveis

Variáveis são espaços de memória reservados para armazenar informações. Estes espaços possuem endereço específico, e representam células elementares que contém um valor que representa uma variável (Sebesta, 2000). O tipo de informação que será armazenada em uma variável deve ser previamente definida na construção do algoritmo.

Existem vários tipos de dados e os mesmos podem ser definidos de maneira distintas nas diversas linguagens de programação existentes. Segundo Wirth (1989) os tipos de dados mais comuns utilizados em algoritmos são:

| Tipo | Exemplo |
|---|---|
| Inteiro | 2 – 50 |
| Real | 2,762 – 100,09 |
| Caracter | "A" - "4" |
| Texto (*String*) | "Algoritmo" - "X" |
| Boolean | Verdadeiro ou Falso |

**Quadro 1 – Tipos de dados**

O tipo do dado é associado a um nome que representará a variável. Este nome é formado por uma letra ou então por uma palavra que signifique o conteúdo armazenado. Não se permite o uso de caracteres especiais (acentos), espaços em branco ou de qualquer outro caractere, que não seja letra ou dígito, na formação do nome da variável (Farrer, 1999).

São alguns exemplos válidos de variáveis:

Numero: inteiro

Nota: real

Nome: texto

Situação: boolean

**2.4 Estruturas**

As estruturas servem para conduzir o fluxo dos dados nos algoritmos, através de testes e condições. Elas diferem umas das outras pela disposição ou manipulação de seus dados ou variáveis. A disposição dos dados em uma estrutura obedece a condições preestabelecidas e caracteriza a estrutura (Szwarcfiter & Markezon, 1994).

Estão divididas em estruturas de decisão e de controle (Sebesta, 2000; Salvetti & Barbosa,1998). As de decisão testam condições que tomam caminhos específicos sem retorno. Elas optam entre duas ou mais possibilidades. As de controle ou de repetição permitem executar mais de uma vez um determinado

número de comandos.

As estruturas de controle servem para executar uma instrução ou um conjunto de instruções repetidamente dependendo da condição determinada no algoritmo. O processo de repetição é conhecido como laço ou *loop* (Salvetti & Barbosa, 1998).

### 2.4.1 Estrutura SE-ENTÃO-SENÃO

É uma estrutura de decisão. Inicia com a palavra especial "SE", seguida pela condição a ser testada. A alternativa a ser tomada se a condição for verdadeira é precedida pela palavra especial "ENTÃO". A outra alternativa a ser tomada se a condição é falsa é precedida pela palavra especial "SENÃO".

Salvetti & Barbosa (1998) e Manber (1989) apresentam esta estrutura na seguinte forma:

```
SE <<condição>> ENTÃO
        <<comandos 1>>
        <<comandos N>>
SENÃO
        <<comandos 1>>
        <comandos N>>
FIMSE
```

Um exemplo do uso desta estrutura está descrito a seguir:

Se João for maior de 18 anos deve ir votar, se ele não for pode ficar em casa. A condição: João maior que 18 anos determina uma ação para João, a de ir

votar. A representação em uma estrutura de condição é:

```
SE idade de João > 18 ENTÃO
        João deve votar
SENÃO
        João pode ficar em casa
FIMSE
```

### 2.4.2 Estrutura ESCOLHA-CASO

É uma estrutura de decisão. É utilizada quando existem diversas opções a serem seguidas dependendo do que o usuário ou o programa solicite. Segundo Ziviani (1999) e Salvetti & Barbosa (1998) esta estrutura é apresentada na seguinte forma:

```
CASO <<condição>> FAÇA
        opção 1: <<comandos 1>>
        opção N: <<comandos N>>
SENÃO
        <<comandos 1>>
        <<comandos N>>
FIMCASO
```

O exemplo a seguir facilita a compreensão desta estrutura:

```
CASO opção ENTÃO
        1: Saldo
        2: Extrato
        3: Depósito
        4: Sair
SENÃO
        Inválida
FIMCASO
```

Lê-se da seguinte maneira: caso a opção seja 1 execute o saldo, caso seja 2 execute o extrato, caso seja 3 execute o depósito, caso seja 4 saia e se a opção não for nenhuma das opções válidas a escolha será inválida. A estrutura ESCOLHA-CASO também pode ser representada pela estrutura SE-ENTÃO-

SENÃO alinhadas, mas é inviável utilizar pois o código ficaria de difícil leitura:

```
SE opção = 1 ENTÃO
        Saldo
SENÃO
        SE opção = 2 ENTÃO
                Extrato
        SENÃO
                SE opção = 3 ENTÃO
                        Deposito
                SENÃO
                        Sair
                FIMSE
        FIMSE
FIMSE
```

O exemplo acima evidencia o aumento da complexidade da leitura se fosse utilizado a estrutura SE-ENTÃO-SENÃO.

### 2.4.3 Estrutura ENQUANTO-FAÇA

Esta estrutura de repetição permite que enquanto uma determinada condição for verdadeira ou válida os comandos pertencentes a estrutura são executados. O término da execução dos comandos está vinculado a condição determinada se tornar falsa. A sintaxe a seguir representa a estrutura ENQUANTO-FAÇA, defina por Sebesta (2000) e Terada & Setzer (1992):

```
ENQUANTO <<condição>> FAÇA
        <<comandos 1>>
        <<comandos 2>>
        <<comandos N>>
FIMENQUANTO
```

O exemplo abaixo utiliza a estrutura ENQUANTO-FAÇA e escreve na tela do computador os números inteiros de 1 até 200.

```
contador := 1
ENQUANTO contador <= 200 FAÇA
        escreva (contador)
        contador := contador + 1
FIMENQUANTO
```

A variável definida como contador será inicializada com o valor 1 e acrescida de uma unidade toda vez que escrever o valor na tela até chegar no valor 200, quando então o laço será finalizado pois a condição estabelecida se tornará falsa. O nome dado a variável que controla a execução do laço é variável de controle (Terada & Setzer, 1992; Guimarães, 1985) .

### 2.4.4 Estrutura PARA-FAÇA

É uma estrutura de controle. É uma simplificação da estrutura ENQUANTO-FAÇA, pois não necessita do incremento da variável de controle, que será automaticamente incrementada na definição da estrutura. A sintaxe abaixo representa a estrutura PARA-FAÇA, segundo Ziviani (1999):

```
PARA <<variável>>:=<<valor inicial>> ATE <<valor final>> FAÇA
        <<comandos 1>>
        <<comandos N>>
FIMPARA
```

O incremento definido na estrutura acima é por padrão uma unidade. O exemplo utilizado no item anterior ficaria da seguinte forma utilizando a estrutura FAÇA-PARA:

```
PARA contador := 1 ATÉ 200  FAÇA
        escreva (contador)
FIMPARA
```

Nesta estrutura a variável contador é inicializada com valor 1 e é acrescida de uma unidade até o valor 200. A forma de inicialização e a taxa de acréscimo já estão definidas na própria estrutura e não necessitam de atribuições em linhas de comandos individuais. Esta estrutura em comparação a ENQUANTO-FAÇA é mais simplificada, porém executa o laço do valor inicial até o final sem interrupção. Caso haja a necessidade de, por algum motivo, o laço seja interrompido a estrutura mais adequada é a ENQUANTO-FAÇA (Guimarães, 1985).

### 2.4.5 Estrutura REPITA-ATÉ

É uma estrutura de controle. Assim como a estrutura ENQUANTO-FAÇA que é usada para repetir diversas vezes uma ou mais instruções, a estrutura REPITA-ATÉ também pode ser aplicada para esta finalidade. A diferença está na validação ou condição, que nesta estrutura, REPITA-ATÉ, é no final, fazendo com que os comandos internos a estrutura sejam executados, pelo menos, uma vez. Sebesta (2000), Salvetti & Barbosa (1998) e Terada & Setzer (1992) apresentam esta estrutura na seguinte forma:

```
REPITA
        <<comandos 1>>
        <<comandos 2>>
        <<comandos N>>
ATE <<condição>>
```

Possui a mesma característica da estrutura ENQUANTO-FAÇA que pode ter a execução do laço interrompida. O exemplo utilizado nos itens anteriores ficaria

desta maneira utilizando a estrutura REPITA-ATÉ:

```
contador := 1
REPITA
        escreva (contador)
        contador := contador + 1
ATÉ contador = 201
```

A variável definida como contador recebe o valor inicial 1. Este valor é escrito na tela do computador pelo comando escreva e em seguida seu valor é acrescido de uma unidade, quando então será feito o teste da condição para a execução do laço. Neste caso se a condição fosse falsa o comando de escrever na tela e o de acrescer a variável de controle já teriam sido executados, pelo menos, uma vez.

## 2.5 Considerações Finais

A programação está apoiada sobre estruturas de decisão e controle. A combinação e o alinhamento entre elas formam a estrutura de um programa ou *software*, que é transformação do algoritmo em uma linguagem comercial, industrial ou científica.

A habilidade de abstrair um problema e encontrar a estrutura de programação adequada para resolvê-lo é a habilidade desenvolvida na disciplina de Algoritmos e Programação. As estruturas descritas neste capítulo estão em consonância com o objetivo da disciplina e com o seu conteúdo programático (anexo 1).

A perícia adquirida em determinar a estrutura ou a combinação certa entre elas para que o problema seja resolvido é o principal objetivo da disciplina de programação, e o modelo proposto nesta pesquisa vem como mais uma solução, para a dificuldade de aprendizado em algoritmos estruturados.

# 3 INTELIGÊNCIA ARTIFICIAL

Neste capítulo encontram-se a revisão bibliográfica sobre Inteligência Artificial (IA), Inteligência Artificial aplicada a Educação (IAED), STI, Formas de Representação do Conhecimento e exemplos de STI aplicados ao ensino da programação estruturada. Um exemplo de STI relacionado ao modelo proposto é apresentado no final deste.

O principal objetivo deste capítulo é trazer um panorama conceitual sobre as definições mais relevantes e significativas utilizadas em IA, IAED, STI e Formas de Representação do Conhecimento. Os exemplos ilustrados de STI aplicados ao ensino da programação estruturada fundamentam o modelo proposto e correlacionam este aos STI já desenvolvidos e aplicados na área de programação.

## 3.1 Inteligência Artificial

Etimologicamente a palavra inteligência vem do latim *inter* (entre) e *legere* (escolher), ou seja, inteligência significa escolher entre uma coisa e outra. Inteligência é a habilidade de realizar de forma eficiente uma determinada tarefa.

A palavra artificial vem do latim *artificiale,* significa algo não natural, isto é, produzido pelo homem. Portanto, inteligência artificial é um tipo de inteligência

produzida pelo homem para dotar as máquinas de algum tipo de habilidade que simula a inteligência do homem (Rabuske, 1995).

Ao iniciar o estudo de IA, uma das maiores dificuldades é tentar delimitar seu campo de estudo, tendo em vista sua vasta abrangência. Segundo Rich (1988) a IA busca entender a mente humana e imitar seu comportamento, levantando as seguintes questões: como ocorre o pensar; como o homem extrai conhecimentos do mundo; como a memória, os sentidos e a linguagem ajudam no desenvolvimento da inteligência; como surgem as idéias; como a mente processa informações e tira conclusões decidindo por uma coisa ao invés de outra. Essas são algumas perguntas que a IA precisa responder para simular o raciocínio humano e implementar aspectos da inteligência.

O histórico da IA está diretamente interligado com o desenvolvimento dos computadores. O termo IA, foi oficialmente, apresentado na conferência em *Dartmouth College*, em *New Hampshire*, nos Estados Unidos, em 1956 (Bittencourt, 1998). Deste encontro participaram grandes pesquisadores da área, como Allen Newell, Herbert Simon, Marvin Minsky e John McCarthy. No mesmo período desta conferência Allen Newell e Herbert Simon tentaram construir um sistema que manipulava símbolos, ao invés dos baseados em números, o que gerou inúmeros estudos para incorporar inteligência às máquinas. Isto se refletiu em enormes mudanças na forma de conceber e utilizar a tecnologia para o ensino e para a aprendizagem.

Diante do surgimento formal desta tão complexa concepção, inúmeras dúvidas e desafios foram colocados perante os pesquisadores, muitos dos quais permanecem até hoje e continuam sendo objeto de continuados estudos. Um dos aspectos que merece consideração especial é o que se refere à própria definição de IA, que para os seus pesquisadores permanece sem uma única interpretação, a qual é feita normalmente em função dos objetivos e metas pretendidos com a utilização deste campo de estudos.

Para Winston (1987) inteligência artificial é "o estudo de conceitos que permitem aos computadores serem inteligentes". Segundo Charniac (apud Barreto, 1998, p. 19) inteligência artificial é "o estudo das faculdades mentais com o uso de modelos computacionais". Rich & Knight (1994) definem inteligência artificial como "o estudo de fazer os computadores realizarem coisas que, no momento, as pessoas fazem melhor".

Para Bittencourt (1998) existem duas linhas de pesquisa em IA: a conexionista e a simbólica. A primeira visa à modelagem da inteligência humana simulando componentes do cérebro. A segunda estabelece uma manipulação simbólica entre os fatos de um domínio. Estas diferentes correntes de pensamentos em IA têm estudado formas de estabelecer comportamentos inteligentes nas máquinas, sendo o grande desafio a tentativa de sintetizar o pensamento de como fazer as máquinas compreenderem as coisas.

A IA fornece métodos e técnicas para o desenvolvimento de programas que simulam nas máquinas comportamentos inteligentes, isto é tornam os computadores capazes de pensar e tomar decisões. Por isso, as técnicas de IA necessitam de uma grande quantidade de conhecimentos e de mecanismos de manipulação de símbolos. Esses conhecimentos devem ter a possibilidade de representação, modificação e ampliação.

A expressão comportamentos inteligentes traz à mente a idéia de máquinas capazes de pensar do mesmo modo que o ser humano, no entanto, para criar uma máquina inteligente não é necessário que ela tenha o mesmo nível de inteligência do homem (Rabuske, 1995). Atualmente já existem máquinas que utilizam aspectos da inteligência humana para realizar tarefas e, no entanto, estão longe de serem comparadas com o nível de inteligência do homem.

Para Rich & Knight (1994) o importante é o aspecto funcional ou prático da simulação da inteligência, por exemplo, um avião voa de maneira similar ao pássaro e não de forma igual, no entanto, isto não invalida ou diminui sua aplicabilidade e a contribuição que o mesmo tem dado para o desenvolvimento da humanidade.

A pesquisa em IA evoluiu muito, mas ainda encontra muitas limitações, pois o objetivo de conferir inteligência aos sistemas é construir uma máquina que venha imitar ou exceder as capacidades mentais humanas, incluindo nestas

capacidades, o raciocínio, a compreensão, a imaginação, a criatividade e as emoções, pois hoje, encontram-se máquinas que imitam áreas específicas e refinadas da atividade mental humana, como por exemplo, os computadores que jogam xadrez.

### 3.2 Inteligência Artificial aplicada na Educação

A evolução das técnicas de IA e das pesquisas no campo das ciências cognitivas, aumentou o grau de inteligência dos sistemas educacionais e antigas dificuldades estão sendo aos poucos superadas (Chaiben, 1998).

Segundo Chaiben (1998) uma das principais motivações para as pesquisas em IAED é o desenvolvimento de princípios pelos quais os ambientes de aprendizagem computacionais possam ser concebidos como lugares onde os estudantes possam ter experiências de aprendizagem individualizadas, isto é, experiências que sejam fundamentais e benéficas para eles, sem importar suas diferenças individuais, experiências anteriores, ou outras situações cognitivas.

Desta forma sistemas que utilizam técnicas de IA podem personalizar o ensino, compatibilizando a apresentação dos conteúdos com o nível de conhecimento do estudante e com o seu índice de aprendizagem.

Todo sistema que tenha como objetivo principal a função de ensinar, deve incorporar princípios de IA (Giraffa, 1996). Os sistemas de IA armazenam e

tratam com dados, adquirem, representam e manipulam conhecimentos, deduzindo e inferindo novos conhecimentos. Estes sistemas estão classificados, segundo Viccari (1996) em: CAI (*Computer Assisted Instruction*), ICAI (*Inteligent Computer Assisted Instruction*) e STI (Sistemas Tutores Inteligentes).

O CAI (*Computer Assisted Instruction*) ou Instrução Assistida por Computador, utiliza a instrução programada, cujo método educacional, influenciado pela Teoria Comportamentalista de Skinner, é a forma expositiva centrada no professor, e onde o aluno deve compreender a lição e depois responder as questões para reforçar sua aprendizagem.

ICAI (*Inteligent Computer Assisted Instruction*) ou Instrução Inteligente Assistida por Computador, se desenvolveram após pesquisas, a partir do CAI e da evolução da IA, bem como da ciência cognitiva, onde houve um aumento da inteligência dos sistemas educacionais. A possibilidade de um ambiente de aprendizagem computacional onde o aluno pode ter uma aprendizagem individualizada, personalizando a instrução através da modelagem do estudante, onde a forma de descoberta é centrada no aprendiz e os diálogos tutoriais são determinados pelo conhecimento conceitual e pelo comportamento da aprendizagem, motivam sobremaneira todos os pesquisadores, desenvolvendo a aplicação da IAED.

Os STI são sistemas educacionais que englobam a IA, chamados de inteligentes e voltados para construção de outros sistemas para a área educacional. Segundo Eberspächer (1998) os STI buscam potencializar a informática educativa através da aplicação de técnicas de IA junto aos programas de computador.

O papel da IA nos programas computacionais educativos pode ser resumido através do destaque de três alternativas de recursos:

- a possibilidade de modelar o conhecimento;
- a capacidade do sistema em resolver problemas que o aprendiz tem que resolver;
- a viabilidade do sistema em conduzir as interações.

A IA dentro da educação possibilita o desenvolvimento de *softwares* que envolvem o raciocínio humano, imitando-o e realizando interferências. Chaiben (1998) afirma que ao simular a inteligência humana através do *software*, as possibilidades de ensino e aprendizagem estarão aumentadas em centenas de vezes.

### 3.3 Sistemas Tutores Inteligentes

Sistemas Tutores Inteligentes são *softwares* de propósito educacional que utilizam técnicas de IA para aquisição e representação do conhecimento. Os STI são um campo de pesquisa e desenvolvimento interdisciplinar e, segundo

Kearsley (1987), os domínios envolvidos neste campo de pesquisa são:

- a Ciência da Computação, com técnicas de IA;
- A Psicologia, com os aspectos cognitivos; e
- os segmentos de Educação e Treinamento, com a aplicação da informática educativa.

A aquisição de conhecimento tem sido considerada um obstáculo para o desenvolvimento de sistemas inteligentes, especialmente para o desenvolvimento do modelo do especialista do STI, onde para se ter eficiência é necessário adquirir e representar uma grande quantidade de conhecimento de vários especialistas em determinado domínio.

Embora derivados dos CAI, os STI oferecem a vantagem sobre estes por simularem o processo do pensamento humano dentro de um determinado domínio. Segundo Dillenbourg (1994) o termo inteligente se refere tanto a técnica usada quanto ao desempenho do sistema, ou seja, aquilo a que o sistema se propõe a fazer.

Para Giraffa (1998, p. 5)

> "o objetivo fundamental dos STI é proporcionar uma instrução adaptada ao aluno, tanto em conteúdo como na forma, superando desta maneira alguns dos problemas mais cruciais do *software* educativo na atualidade. Os STI se comportariam de forma mais próxima a um professor humano ou um comportamento mais próximo possível disto. Porém, a realidade está muito distante de alcançar tais propósitos. Existem muitas razões para que isto ocorra: temos limitações a nível de *hardware* e *software* que não nos permitem colocar dispositivos que possam trabalhar com aspectos relativos aos sentidos do olfato, tato e visão. Um professor humano pode e leva em consideração

> estes estímulos para poder organizar seu trabalho junto ao aluno e, além disto, utiliza as saídas deste sentidos para fins de *feedback* do aluno. O fato é que desconhecemos a maneira com que nós humanos efetivamente processamos informação dentro de nosso cérebro".

### 3.3.1 Estrutura de um STI

A maioria dos STI desenvolvidos possuem certas características em comum, sendo que suas implementações seguem uma arquitetura básica constituída de quatro módulos. Segundo Kaplan & Rock (1995) e Giraffa (1998), os STI são estruturados em:

- Modelo do Especialista;
- Modelo do Estudante;
- Modelo Pedagógico;
- Modelo de Interface.

A figura 1 representa a estrutura básica de um STI.

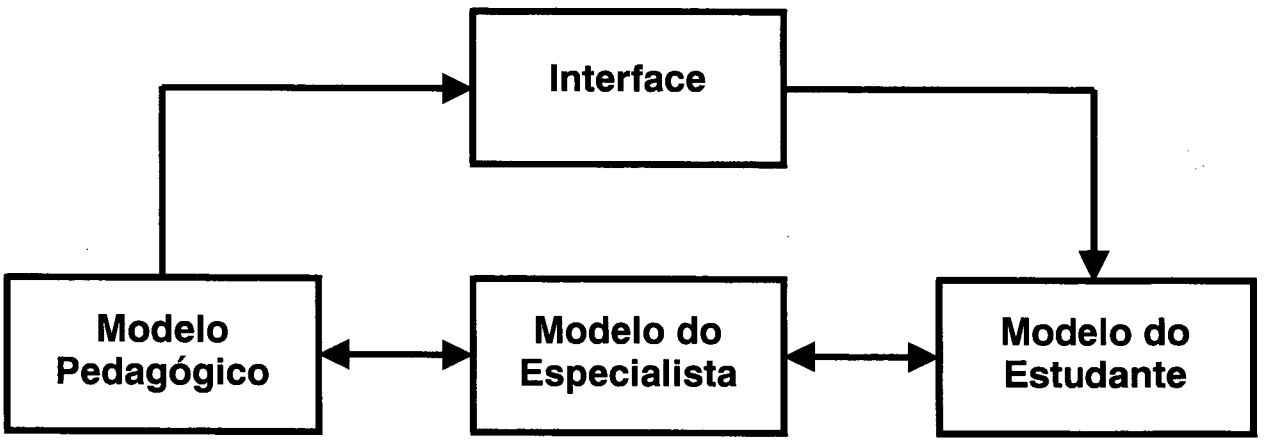



**Figura 1** - Diagrama em blocos de um STI básico.

3.3.1.1 Modelo do Especialista

O Modelo do Especialista ou conhecimento de domínio (Eberspächer, 1998) descreve o conhecimento de um especialista na área de domínio do sistema, servindo como base para a construção do Modelo do Estudante.

Este modelo contém o conhecimento sobre o domínio que se deseja ensinar ao estudante. Segundo Giraffa (1996) vários modelos de representação de conhecimento podem ser usados aqui: redes semânticas, *frames*, *scripts*, regras de produção, entre outras. A escolha deve recair sobre aquele método que melhor e mais facilmente atenda os requisitos de representação e manipulação do raciocínio.

Esta base de conhecimento contém os elementos necessários para que o estudante aprenda o conhecimento e os procedimentos necessários para que possa utilizar na resolução de problemas.

Woolf (1988) e Giraffa (1998) citam que nos casos em que o domínio a ser representado é de natureza descritiva e teórica (por exemplo, geografia ou física), a representação do conhecimento mais adequada é declarativa (redes semânticas ou *Frames*). Já nos casos em que o domínio é orientado a uma tarefa (por exemplo, programação estruturada ou Pascal), a representação tende a ser procedural, onde se aplicam as regras de produção.

3.3.1.2 Modelo do Estudante

O Modelo do Estudante é a representação do conhecimento do estudante e dos seus erros, mapeando quais informações do professor que já foram assimiladas. Ele contém uma representação do estado do conhecimento do aluno no momento que interage com o STI (Giraffa, 1996).

É neste modelo que reside a chave para um ensino personalizado e inteligente. A dimensão mais significativa de um STI é sua capacidade de modelar o conhecimento do estudante (Jonassen & Wang, 1993).

Este modelo deve ser dinâmico, contendo o conhecimento e as capacidades do estudante bem como seu comportamento de aprendizagem passado. Segundo Costa (1996) e Eberspächer (1998) o modelo do aluno pode ser representado, apoiando-se em alguns modelos de descrição como:

- Modelo diferencial: neste modelo a resposta do aluno é comparada com a da base de conhecimento.
- Modelo de Overlay ou superposição: o conhecimento do aluno é representado como um subconjunto da base de conhecimento do STI.
- Modelo de Perturbação: este modelo também relaciona o modelo do aluno com a base de conhecimento do domínio, porém assume que os erros do aluno são decorrentes da concepção errônea de algum conceito ou ausência dele.
- Modelo de Crenças: consiste em um conjunto de crenças que refletem que

pensamos o quanto o estudante entende sobre determinado conceito.

3.3.1.3 Modelo Pedagógico

O Modelo Pedagógico, também chamado de Módulo Instrucional (Eberspächer, 1998), representa os métodos e técnicas didáticas utilizadas no processo da comunicação de conhecimento. Executa o diagnóstico do estudante, decide quais as estratégias de ensino serão utilizadas e determina qual a informação que será apresentada.

Sua principal função é a de gerenciar a seqüência das instruções, monitorando o comportamento e a performance do estudante de maneira a auxiliá-lo no processo de aprendizagem.

Neste modelo o STI pode mudar de estratégia pedagógica de acordo com os resultados que o estudante vem apresentando. Wenger (1987, p.45) enfatiza que

> "as decisões pedagógicas são tomadas no contexto de um ambiente educacional que determina o grau de controle sobre a atividade e sobre a interação possuídos respectivamente pelo sistema tutorial e pelo estudante".

Portanto, um processo de aprendizagem depende de diversos fatores e o STI deve observar para não destruir a motivação pessoal do estudante ou o senso de descobrimento. Breuker (apud Giraffa, 1996, p. 32) cita que a estratégia de ensino adotada deve definir:

- "Quando ocorrerá a interrupção do STI sobre o raciocínio ou aprendizagem

do aluno;
- O que deverá ser abordado, considerando a seleção dos tópicos apresentados e sua ordenação;
- Como transmitir os conhecimentos inseridos dentro de cada tópico a ser explorado. As teorias pedagógicas existentes trabalham como o conhecimento deve ser transmitido".

3.3.1.4 Modelo de Interface

O Modelo de Interface ou, simplesmente, Interface é a forma como a comunicação será realizada com o meio externo ao sistema. Ele apresenta o material instrucional que será armazenado na base de conhecimento (Modelo do Especialista) ao estudante, traduzindo a representação do conhecimento adotada em uma linguagem de interface compreendida pelo estudante.

Na engenharia de *software*, a interface do usuário tem sido a primeira preocupação dos projetistas quando estão discutindo a criação de uma nova aplicação pois, para os usuários, a interface é o próprio sistema (Chaiben, 1998).

Este modelo é fundamental para o sucesso de um STI, já que ele é responsável pela apresentação do material e por receber os eventos e solicitações do usuário. É responsável pelo fluxo de informação de entrada e de saída, transformando as solicitações de entrada em informações que o STI possa compreender, bem como adaptando as de saída para o contexto, mais adequado, do estudante.

Segundo Giraffa (1996) este módulo possui duas principais funções:

- a apresentação do material instrucional;
- a monitoração do progresso do estudante através da recepção da resposta do aluno.

A partir destas funções, Giraffa (1998) cita algumas características do módulo de interface:

- é necessário evitar que o estudante se entedie, ou seja, é preciso riqueza de recursos na apresentação do material instrucional;
- é desejável que haja facilidade para troca da iniciativa do diálogo: o estudante deve poder intervir facilmente no discurso do tutor, e vice-versa;
- o tempo de resposta deve, evidentemente, permanecer dentro de limites aceitáveis;
- a monitoração deve ser realizada o máximo possível em *background*, para não onerar o estudante com questionários excessivos, mas respeitando também a barreira do tempo de resposta.

### 3.4 Formas de Representação do Conhecimento

A representação do conhecimento é o componente fundamental em STI. Os mecanismos utilizados na representação dos conhecimentos determina o comportamento do STI (Viccari, 1996).

De acordo com os formalismos de IA, o conhecimento é codificado através de objetos, atributos, objetivos, ações e é processado através de estruturas e procedimentos. A representação do conhecimento é uma redução, coerente e com o senso comum suficiente, de determinadas circunstâncias que permitirão o computador decidir de forma semelhante a um ser humano (Bittencourt, 1998).

Para representar o desempenho de especialistas humanos, o sistema deve possuir não só um conjunto de informações mas, também, a habilidade de utilizá-las na resolução de problemas de forma criativa, correta e eficaz. Segundo Fialho (1998) esta habilidade representa uma série de palpites e regras intuitivas que o especialista utiliza para resolver os problemas; sua aplicação possibilita, de uma maneira mais econômica, a chegada a soluções aceitáveis, embora nem sempre ótimas.

Um STI precisa conhecer o contexto do fato em estudo e reconhecer os processos que causariam mudanças nos fatos. Para resolver problemas, em alguns casos, é recomendado conhecer tudo sobre o problema e quais as possíveis soluções que se pretende encontrar, juntamente com algumas estratégias para solucionar cada problema. O estudo da representação do conhecimento consiste nos caminhos que podem ser trilhados para codificá-lo em um programa computacional.

Para Chaiben (1998) uma das principais características dos programas de IA é que o sistema é estruturado de modo a separar o código executável dos dados ou conhecimento do sistema. Assim, em IA, o termo conhecimento significa a informação que um programa de computador necessita para que possa comportar-se inteligentemente.

Segundo Fialho (1998) os conhecimentos que representam objetos são modelados por rede semântica, os que representam situações e os acontecimentos, são expressos por esquemas e os que representam ações são modelados por procedimentos.

### 3.4.1 Procedimentos

Segundo Chaiben (1998) a representação procedimental é aquela em que as informações de controle necessárias ao uso do conhecimento estão embutidas no próprio conhecimento, ou seja, a maior parte do conhecimento é representada como procedimentos para a sua utilização.

Nesta representação o conhecimento é expresso em forma de procedimentos ou passos seqüenciais que determinam ações, diagnosticam problemas e encontram soluções. Este tipo de conhecimento é usualmente representado em um conjunto de regras ou árvores de decisão. Dentre as principais vantagens da utilização da representação procedimental incluem-se (Rich, 1988):

- A facilidade em representar o conhecimento de como fazer as coisas;

- A facilidade em representar o conhecimento que não se enquadra dentro de muitos esquemas declarativos simples, como por exemplo, o raciocínio por omissão e o raciocínio probabilístico;
- A facilidade em representar o conhecimento heurístico de como fazer eficientemente as coisas.

O exemplo abaixo representa o conhecimento em forma de procedimento, através das regras de produção:

SE o carro não quer ligar e a bateria está carregada

ENTÃO examine o distribuidor

### 3.4.2 Esquemas

A noção de esquemas foi introduzida para explicar o papel desempenhado pelos conhecimentos na compreensão, na memorização, e na produção de inferências. Os mais significativos são os *scripts* e os *frames*.

#### 3.4.2.1 *Scripts*

Os *scripts* são estruturas de informação que auxiliam a compreensão de situações do comportamento padronizado. Foram propostos por Schank & Abelson em 1977 (apud Barreto, 1998, p. 59) e inspiraram o estudo de sistemas de Raciocínio Baseado em Casos (RBC). Os *scripts* são úteis porque, no mundo real há padrões para a ocorrência de eventos. Contudo, o conceito de um *script* não é compartilhado por todos, já que cada memória compreende

um *script* sobre uma experiência a partir do próprio ponto de vista. Portanto a teoria dos *scripts* não é uma teoria completa. Os *scripts* contém o conhecimento normativo, mas não o conhecimento da experiência.

Um exemplo de script, segundo Schank & Abelson (apud Fialho 1998, p. 218), é o do restaurante e é representado da seguinte forma:

| | |
|---|---|
| Trilha: Sala do Café | Participam: |
| Props: Mesa | S - Cliente |
| Menu | W - Atendente |
| F – Comida | C - Cozinheiro |
| Cheque | M - Recebedor |
| Dinheiro | O - Proprietário |

| | |
|---|---|
| Condições de Entrada: | Resultados: |
| S está com fome | S tem menos dinheiro |
| S tem dinheiro | O tem mais dinheiro |
| S não está com fome | |
| S está satisfeito (opcional) | |

Cena 1: Entrada

S Ptrans S para dentro do restaurante

S Attend olhos para a mesa

S Mbuild onde sentar

S Ptrans S para a mesa

S Move S para a posição sentada

Cena 2: Fazendo o pedido

(menu sobre a mesa) (W traz menu) (S pede pelo menu)

S Ptrans menu to S S Mtrans um sinal to W

W Ptrans W to mesa

S Mtrans 'preciso do menu' to W

W Ptrans W to menu

W Ptrans W to mesa

W Atrans menu to S

S Mtrans a lista de comida to CP (S)

* S Mbuild escolha de F

S Mtrans sinal to W

W Ptrans W to mesa

S Mtrans 'Eu quero F' to W

W Ptrans W to C

W Mtrans (Atrans F) to C

C Mtrans 'não tem F' to W

W Ptrans W to S

W Mtrans 'não tem F' to S

(volta para *) ou

(vai para a CENA 4 para o caminho de não pagamento)

C Do (prepara F script)

vai para a CENA 3

Cena 3: Comendo

C Atrans F to W

W Atrans T to S

S Ingest F

(opcionalmente retorna a Cena 2 e pede mais, ou vai para a Cena 4)

Cena 4: Saindo

S Mtrans to W

(W Atrans cheque to S)

W Move (escreve cheque)

W Ptrans W to S

W Atrans cheque to S

S Atrans gorjeta to W

S Ptrans S to M

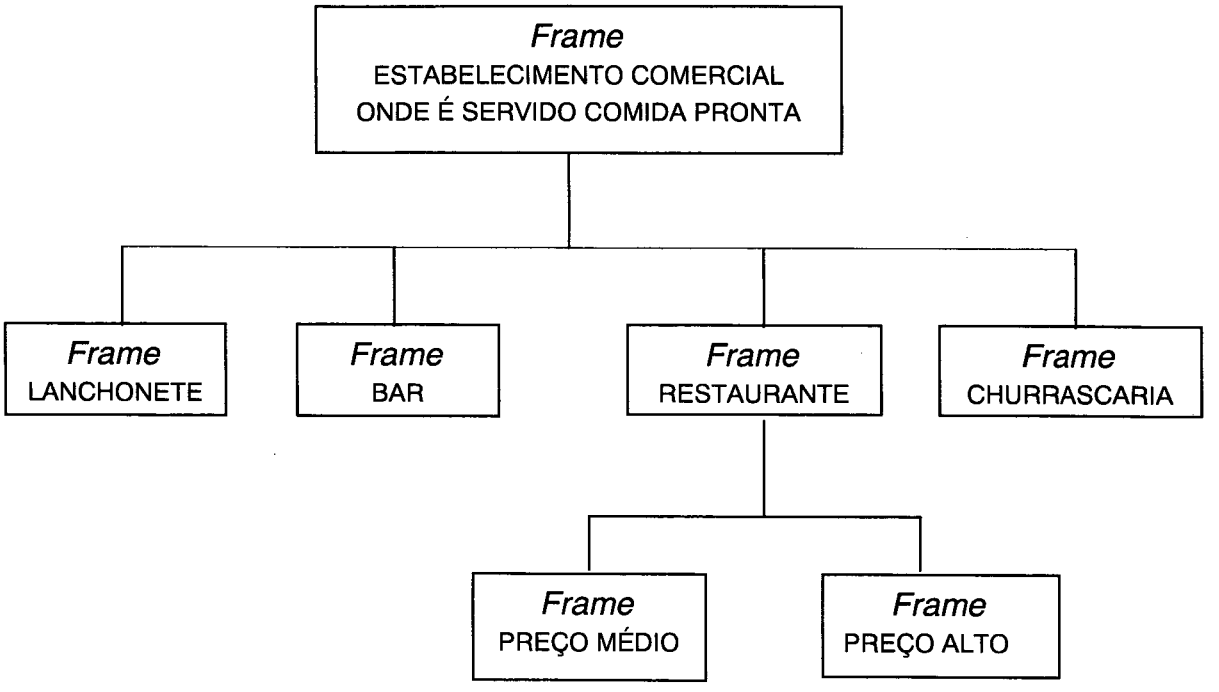
S Atrans dinheiro to M

(caminho de não pagamento) S Ptrans S to fora do restaurante

### 3.4.2.2 *Frame*

Segundo Fialho (1998) *frame* é uma estrutura de dados que representa uma entidade através de suas características e potenciais habilidades. Suas características estão representadas por pares atributo-valor e as potencialidades são representadas por métodos. Um *frame* abstrato (ou *frame* de classe) não tem instâncias, por esta razão seus atributos não são valorados, suas subclasses são ligadas a instâncias da entidade representada por essa classe.

Para exemplificar, será representado o conhecimento necessário a uma situação específica, como "jantar fora" através de *frames*. Para "jantar fora" é necessário um estabelecimento comercial onde é servido comida pronta; sabe-se, no entanto, que existem vários tipos desses estabelecimentos: lanchonetes, bares, restaurantes, churrascarias, entre outros, cada um com características próprias. Levando em conta que os restaurantes ainda podem ser de preço médio ou preço alto, tem-se a seguinte estrutura de *frames*:




Uma sugestão para o *frame* RESTAURANTE:

1. Especialização de estabelecimento comercial onde é servido comida pronta;

2. *Frame*s correlatos: lanchonete, bar, churrascaria

    Se refeições rápidas então lanchonete

    Se bebidas vendidas em balcão então bar

    Se carne servida em rodízio então churrascaria

3. Estilo de comida: brasileira, portuguesa, italiana; default: brasileira

4. Forma de operação: com reserva, sem reserva; default: sem reserva

5. Horário de funcionamento: variável; default: 09:00 h da manhã à 02:00 h da madrugada

6. Forma de pagamento: dinheiro, cheque, cartão

7. Seqüência de eventos: Roteiro jantar no restaurante

8. Quadros especializados: preço médio e preço alto

Se maioria dos pratos custa entre 10 e 50 reais, então preço médio

Se maioria dos pratos custa acima de 50 reais, então preço alto

### 3.4.3 Redes Semânticas

As Redes Semânticas são grafos direcionados ligados por nós para representar objetos e conexões e a relação entre estes. A rede semântica é usada para representar elementos de uma representação tal como uma classe, suas instâncias e suas características. Seus arcos são direcionados e representam as relações entre os atributos.

Chaiben (1998) define rede semântica como uma estrutura de representação do conhecimento definida como um padrão de nodos interconectados por arcos rotulados. As redes deste tipo não só captam as definições dos conceitos mas também, inerentemente, proporcionam ligações com outros conceitos.

Para representar, por exemplo, que "Todo carro é um veículo" temos:

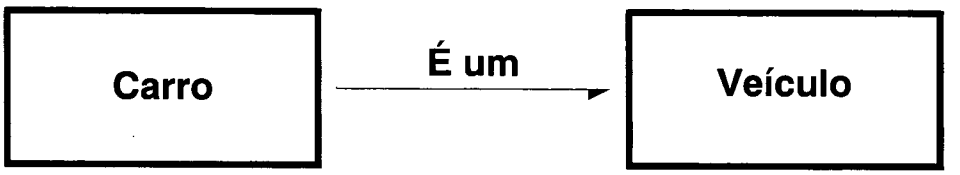



**Figura 2** – Exemplo de Rede Semântica.

Os arcos em geral dependem da espécie de conhecimento que está sendo

representado; por exemplo, é-um e é-parte-de são arcos para representar hierarquias entre objetos. Uma característica chave da representação por rede semântica é que importantes associações podem ser feitas explicitamente e sucintamente. Fatos importantes sobre um objeto ou conceito podem ser deduzidos dos nós aos quais eles estão ligados diretamente, sem uma pesquisa no contexto.

Desta forma, o exemplo a seguir mostra uma rede semântica mais abrangente que representa "Todo carro tem rodas é um veículo é um meio de transporte":

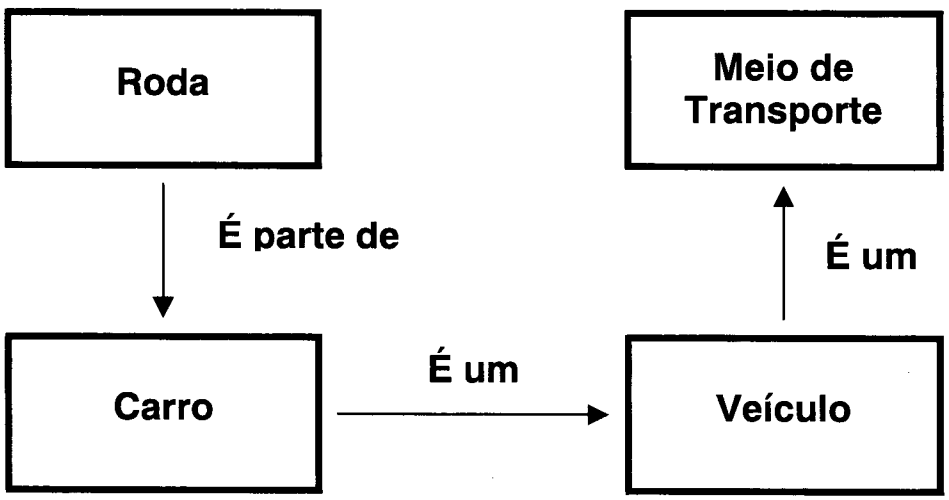



**Figura 3** – Exemplo de Rede Semântica.

### 3.5 Exemplos de STI aplicados a Programação Estruturada

Os STI destinados a iniciantes de programação associam técnicas de IA, ambientação construtivista por exploração livre e recursos de visualização

científica para, no final, criar interfaces de razoável versatilidade para o aluno (Pimentel & Direne, 1997).

Direne & Pimentel (1998) cita alguns exemplos de STI aplicados ao ensino da programação: o Lisp Tutor, GIL, LAURA, BIP, Proust, Spade, Talus e Bridge. Entretanto, estes STI e micromundos enfatizam apenas os aspectos semânticos da solução de um exemplo de programação, independentemente do grau de dificuldade do enunciado do exercício em questão.

O projeto MENO, iniciado no final da década de 70 tinha por objetivo construir um STI para ensinar alunos iniciantes na linguagem Pascal (programação estruturada). Segundo Giraffa (1998), o objetivo deste projeto era diagnosticar erros não sintáticos em pequenos programas e associá-los ao nível de compreensão apresentado pelo aluno. Desta forma era possível identificar qual o conceito que o aluno não aprendeu bem através do tipo de erro que ele cometeu.

A utilização destes STI apontaram a integração de comandos individuais como a principal dificuldade demonstrada por iniciantes que utilizam linguagens de programação convencionais, ou seja, a falta de perícia na utilização das estruturas básicas de programação (Direne & Pimentel, 1998).

Os STI apresentados não consideram a complexidade do enunciado de um

problema de programação, que possui diversas componentes cognitivas que aumentam o grau de dificuldade da resolução do problema. Parte da importância deste procedimento pode ser constatado quando um problema de programação significativamente mais complexo do que o anterior é proposto a um aprendiz. Esta fato pode levar à ocorrência de erros múltiplos, impedindo o sucesso do aluno por longos períodos de tempo (Pimentel & Direne, 1997).

Alguns STI como BALSA II (Brown, 1988) e o TINKER (Liberman, 1984), apresentam, aos alunos, seus enunciados em uma ordem de complexidade extremamente rígida e inflexível, o que aumenta o grau de compreensão do enunciado proposto, dificultando a resolução dos exercícios.

### 3.6 Exemplo de STI relacionado ao Modelo Proposto

Silveira (1996) desenvolveu um STI denominado de ELETROTUTOR, cujo objetivo é auxiliar o ensino de tópicos de eletricidade, Lei de Ohm e temas relacionados, para alunos do terceiro ano do segundo grau e cursos de eletricidade básica.

O conteúdo está dividido em oito unidades estruturadas em um texto base sobre o assunto, um gerador de exemplos e um de exercícios. Os textos são apresentados em telas e os exemplos gerados são estáticos, não permitindo qualquer edição ou modificação por parte do aprendiz.

Os exercícios são do tipo V (verdadeiro) ou F (falso). Se o aluno erra a resposta correta é apresentada antes do novo exercício. Os problemas propostos ao aluno são escolhidos randomicamente a partir de um arquivo de exercícios. Este processo de escolha de problemas é aleatório, considerando a atuação do aluno no exercício anterior.

O sistema faz um diagnóstico inicial, com perguntas, a fim de levantar o perfil do aluno, após isto faz um plano de trabalho que o aluno pode ou não aceitar. Os textos e os exercícios mudam em função da atuação do aluno. O protótipo foi implementado em Arity-Prolog e serviu para avaliar o desempenho de um STI de física e foi utilizado como base para o projeto ELETROTUTOR-II (Silveira, 1998).

### 3.7 Considerações Finais

Diante de tais colocações, a IA é uma tecnologia chave para o *software* do futuro, portanto, pode-se afirmar que o campo de IA tem como objetivo, o contínuo aumento da inteligência do computador, pesquisando, para isto, também os fenômenos da inteligência natural na resolução de problemas, da compreensão de linguagem natural, da visão e da robótica, formas de aquisição e metodologias de representação de conhecimento.

O desenvolvimento científico e tecnológico vem criando nos educadores a necessidade de adotar modelos de ensino que atendam às profundas

modificações que a sociedade do início do novo século passa a exigir, onde a crescente perspectiva de diversificar os espaços educacionais revela um aprendizado sem fronteiras.

A adoção das técnicas de IA aos *softwares* educacionais tornam possível a criação de ambientes de aprendizagem, nos quais as diferenças e experiências individuais dos aprendizes são consideradas (Alves & Simões, 1999).

A modularização em um STI e as pesquisas individualizadas de aprimoramento de cada modelo, destacando a separação entre estratégias de ensino e conhecimento representado proporciona um aumento significativo na interação do aprendiz e o sistema. Os STI estão em aprimoramento. Ainda não cumprem todos os objetivos a que se propuseram resolver.

Chaiben (1998) cita que as razões estão relacionadas aos altos custos de desenvolvimento e a falta de um paradigma estabelecido para descrever o processo de aquisição de conhecimento. Várias teorias foram desenvolvidas, mas nenhuma tem sido aceita como um modelo apropriado de cognição. Outro problema, é a incapacidade de um sistema em gerar um raciocínio pedagógico inteiramente autônomo, o que possibilitaria ao sistema tomar decisões que não tivessem sido antecipadas pelos especialistas. Assim, os projetos são baseados em modelos que podem ou não representar o processo de aquisição de conhecimento.

Desta forma STI atuam como complementos à forma de aprendizagem aplicada, fortalecendo a transmissão dos conhecimentos e possibilitando novas maneiras de acompanhar o aprendiz no processo ensino-aprendizagem. O aprimoramento constante dos STI, e o aperfeiçoamento das técnicas de IA, aliadas ao desenvolvimento de novos *hardwares* e *softwares* conduzem, para um futuro próximo, a criação de STI capazes de, integralmente, se modelar a necessidade de aprendizagem e as características do estudante.

# 4 TEORIAS DE APRENDIZAGEM

O objetivo deste capítulo é descrever as principais correntes do pensamento para os modelos educacionais, com base nas idéias das obras de seus representantes mais relevantes, interrelacionando-as e demonstrando as mais significativas características em relação a aprendizagem, relacionamento professor-aluno, metodologia aplicada e forma de avaliação.

O capítulo inicia com a definição de aprendizagem e na seqüência são descritas três das principais concepções: a empirista, que fundamenta-se em um ser humano que aprende com base nas experiências que vive, a inatista, que refere-se a um ser humano pronto, desde o seu nascimento, onde a aprendizagem apenas lapida o que já existe e a interacionista, que divide-se em o construtivismo de Jean Piaget, onde o ser humano constrói seus conhecimentos seguindo fases pré-determinadas e o sócio-interacionismo de Vygotsky, onde o ser humano aprende interagindo com o ambiente na mesma proporção que este interage com o ser humano.

## 4.1 Considerações Iniciais

A escola é, sem dúvida, o elemento transformador e impulsionador de uma sociedade. Sua principal função, segundo Falcão (1996), é que o aluno aprenda e que o professor oriente a aprendizagem de seus alunos. Na escola o aluno deve possuir espaço para criar, aprender, escutar, criticar, analisar,

desenvolver, construir, pesquisar e principalmente ser orientado a aprimorar seus conhecimentos (Barros, 1998). Mas todas estas atividades só se concretizam com a transformação da informação em conhecimento através da aprendizagem.

A aprendizagem é o objetivo principal a ser buscado por todos os professores. Ela proporciona uma mudança no comportamento do aluno. Novas informações, conceitos, pontos de vista são adquiridos de forma permanente. Esta mesma visão é defendida por Cória-Sabini (1986) quando enfatiza que a aprendizagem está relacionada a um processo de mudança não só de conhecimentos mas também de atitudes, como por exemplo, preferências e preconceitos.

Segundo Hilgard (apud Cória-Sabini, 1986, p. 1) a aprendizagem é "o processo pelo qual uma atividade tem origem ou é modificado pela reação a uma situação encontrada". Nesta visão aprendizagem é uma resposta a um estímulo recebido diante de determinada situação.

Para Falcão (1996, p.20), aprendizagem está relacionada a "uma modificação relativamente duradoura do comportamento, através de treino, experiência e observação".

Estas duas visões de conceitos se relacionam na questão da mudança do

comportamento. Na escola esta mudança proporciona ao aluno a capacidade de inferir novos conhecimentos a partir dos exercícios, da teoria e, principalmente, da prática.

Para compreender melhor o desenvolvimento da aprendizagem, foram desenvolvidas, dentro da psicologia da educação, diferentes teorias, cada uma com o mesmo objetivo: o aprendizado do aluno. Todas as teorias contribuem para o desenvolvimento do aluno, porém de acordo com a concepção de ser humano, cada uma exigirá do educador uma atuação (Barros, 1996).

Para Barros (1996) e Davis (1994) as concepções dividem-se em três: o empirismo, o inatismo e o interacionismo.

### 4.2 Empirismo

A concepção empirista ou ambientalista, também citada por Mizukami (1986) como comportamentalista ou behaviorista, acredita que o desenvolvimento do ser humano está relacionado diretamente com o seu ambiente. Ele é um ser maleável que se molda em função das condições presentes no meio em que se encontra. Devido esse papel exercido pelo ambiente sobre o homem, não existe, nesta corrente, a possibilidade de liberdade individual, de escolha e de autenticidade.

Para Barros (1996) esta concepção também é conhecida como objetivista, pois

fatores internos como a maturação biológica, a inteligência, aptidões, vontade e sentimentos não são considerados, apenas os fatores externos proporcionam o desenvolvimento da pessoa. Esta visão objetivista não emprega o conceito de motivação para fazer o aluno responder melhor em uma determinada situação, pelo contrário, a resposta que o aluno dará será conseqüência de um esquema de reforço (Cória-Sabini, 1986).

Para esta corrente quando o homem nasce sua mente é considerada uma tábula rasa, uma página em branco, um espaço pronto para receber as experiências que serão vividas no meio em que se encontra. Nesta concepção a experiência sensorial será a fonte do conhecimento (Davis, 1994). E este conhecimento é uma descoberta e é nova para o indivíduo que a faz (Mizukami, 1986).

Nesta abordagem a aprendizagem é conceituada como uma mudança no comportamento, resultante de estímulos recebidos do ambiente (Barros, 1996). Esta visão comportamentalista enfatiza que um determinado comportamento poderá ser mantido ou tornado mais freqüente se for seguido pela apresentação de estímulos reforçadores ou efeitos agradáveis. Em contra partida, um comportamento poderá ser eliminado se estes estímulos forem retirados, ou ainda, se for aplicada uma punição.

Burrhus Frederic Skinner pode ser considerado o representante mais

significativo desta corrente e o mais difundido no Brasil (Misukami, 1986). Para ele os estímulos, elementos presentes no ambiente, que provocam um comportamento de efeito agradável são chamados de estímulos reforçadores positivos, e os que ocasionam um comportamento desagradável são identificados como estímulos reforçadores negativos.

Dentro do comportamento defendido por Skinner o papel do ambiente é muito mais importante que a maturação biológica. Nesta corrente educar é estabelecer condicionamentos. Por exemplo, se após arrumar os seus brinquedos (comportamento), a criança ouvir elogios de sua mãe (conseqüência positiva), ela procurará deixar os brinquedos arrumados por mais vezes, porque estabeleceu uma associação entre esse comportamento e aquele de sua mãe. Neste exemplo temos os estímulos reforçadores positivos.

Um exemplo de reforçadores negativos, é quando uma criança quebra uma vidraça jogando bola (comportamento), se ela for obrigada a pagar pelo estrago (conseqüência negativa), ela terá mais cuidado toda vez que for jogar bola, quem sabe esta conseqüência faça esta criança brincar com bola em outro lugar.

Há momentos em que o comportamento é inadequado e deseja-se eliminá-lo. Este procedimento é chamado extinção. Para Davis (1994) um comportamento pode ser extinto quando quebramos o elo que se estabeleceu entre o

comportamento visto como indesejável e as conseqüências do mesmo. Se, por exemplo, um aluno fizer bagunça em sala de aula para chamar a atenção do professor, mas este não der mostras de que está notando o comportamento do aluno, é provável que o aluno pare de fazer bagunça. Este exemplo mostra que o comportamento do aluno foi extinto porque deixou de promover o aparecimento de determinadas conseqüências, neste caso a atenção do professor.

### 4.2.1 Aprendizagem

Com base nas idéias desta concepção cabe ao professor observar e condicionar as respostas do aluno de acordo com a situação corrente. O aluno é passivo, não reage de forma diferente senão aquela manipulada a acontecer. Este condicionamento imposto ao aluno é a base dos estudos desenvolvidos por Skinner. Mizukami (1986) afirma que, "segundo Skinner, cada parte do comportamento é uma função de alguma condição que é descritível em termos físicos, da mesma forma que o comportamento". A aprendizagem provoca uma mudança comportamental como resultado de uma prática reforçada.

A aprendizagem, na visão ambientalista, necessita que se considere o estado fisiológico e psicológico de quem está aprendendo. Devem ser considerados a natureza dos estímulos presentes na situação e o tipo de resposta que se espera obter (Davis, 1994). Para Mizukami (1986) e Falcão (1996) os comportamentos desejados dos alunos são instalados e mantidos por

condicionantes e reforçadores arbitrários, tais como: elogios, graus, notas altas, prêmios, reconhecimento do professor, entre outros, os quais por sua vez, estão associados com uma outra classe de reforçadores mais remotos e generalizados, tais como: diploma, ascensão social e monetária, *status*, prestígio da profissão, entre outros.

Ensinar, nesta abordagem, é planejar as contigências de reforços sob as quais os estudantes aprendem. Este planejamento é de responsabilidade do professor que também assegura que haverá a aquisição do comportamento. O planejamento está baseado na aplicação de métodos científicos de investigação e de elaboração de técnicas e intervenções que objetivam as mudanças comportamentais. Este planejamento inclui os objetivos instrucionais e operacionais que estipulam a seqüência de atividades que levarão ao objetivo proposto e a especificação dos reforçadores, positivos e negativos, que serão utilizados. Alguns elementos mínimos são exigidos para a consecução de um sistema instrucional, como: o aluno, um objetivo de aprendizagem e um plano para alcançar o objetivo proposto.

Para Skinner, segundo Mizukami (1986), o ensino corresponde ao arranjo ou a disposição de contigências para uma aprendizagem eficaz. Esse arranjo, por sua vez, depende de elementos observáveis na presença dos quais o comportamento ocorre: um evento antecedente, uma resposta, um evento conseqüente (reforço) e fatores contextuais.

Na concepção ambientalista, a eficiência na elaboração e utilização dos modelos de ensino, estão relacionadas as habilidades do planejamento desenvolvido pelo professor. Na visão de Skinner é possível programar o ensino de qualquer disciplina, tanto quanto o de qualquer comportamento, bastando para isso que o professor planeje o objetivo final a ser alcançado e as atividades necessárias para que isto ocorra.

A ênfase dada é a figura do professor, nesta abordagem, pois ele possui a responsabilidade de modificar o comportamento dos alunos e promover a aprendizagem. Ele planeja, organiza e executa as situações de aprendizagem. Isto exige do professor profundos conhecimentos dos fatores a serem considerados numa programação de ensino.

A aprendizagem será garantida pela programação desenvolvida pelo professor e pelos recursos tecnológicos que ele utilizará. Davis (1994) propõe que as situações de ensino devam ser bem estruturadas e planejadas previamente, recorrendo-se, sempre que possível, à presença de computadores, televisão e outros recursos audiovisuais.

### 4.2.2 Relacionamento Professor-Aluno

Na abordagem empirista o professor é o centro e responsável pela administração das condições que proporcionam a transmissão dos conhecimentos, não importando as relações afetivas e pessoais envolvidas no

processo (Silva, 2000). Ele também planeja e desenvolve o sistema de ensino-aprendizagem, de forma tal que o desempenho do aluno seja maximizado em todos os fatores, como: tempo, esforços e custos (Mizukami, 1986).

No planejamento o professor deve considerar o estado comportamental inicial de seus alunos e conduzir as atividades intermediárias de maneira que todos apresentem o comportamento final esperado. O professor é considerado como planejador e analista de contigências. Deverá dispor estas contigências em relação às respostas desejadas.

Como centro da transmissão dos conhecimentos, também é de sua responsabilidade selecionar o conteúdo e a forma com que estes serão apresentado aos alunos.

Ao aluno cabe receber, escutar e repetir as informações recebidas, tantas vezes quanto forem necessárias, até acumular em sua mente o conteúdo transmitido pelo professor (Silva, 2000). Ele assimila o que o professor determinar. Ele é passivo nesta relação e não interage com o professor. Não participa e nem interfere no planejamento e desenvolvimento da aula.

### 4.2.3 Metodologia

Esta concepção trouxe ao professor a responsabilidade de definir os objetivos e confeccionar o planejamento das aulas. Mas para os objetivos serem

realmente atingidos é necessário que o professor determine as estratégias e quais as tecnologias de ensino que serão aplicadas.

Mizukami (1986) cita três formas de ensino que são utilizadas dentro desta corrente:

- ensino individualizado;
- ensino baseado na competência;
- ensino programado.

4.2.3.1 Ensino individualizado

A individualização do ensino implica em maior envolvimento do aluno e controle do professor nos elementos que especifiquem o domínio de uma determinada habilidade. O professor poderá ter melhor controle do ritmo de mudança comportamental do aluno, acompanhando-o na sua evolução passo-a-passo.

Este tipo de ensino maximiza a aprendizagem, desempenho e desenvolvimento do aluno. Permite ao professor o acompanhamento dos objetivos propostos e assegura que um maior número possível atinja altos níveis de desempenho. Uma aplicação desta estratégia de ensino pode ser direcionada as pessoas com algum tipo de deficiência física ou mental.

4.2.3.2 Ensino baseado na competência

Esta estratégia de ensino utiliza o módulo instrucional como material de ensino

(Mizukami, 1986). Este ensino é caracterizado por:

- especificação dos objetivos em termos comportamentais;
- especificação dos meios para se alcançar o desempenho esperado;
- conhecimento do público que receberá o ensino;
- critérios e formas para atingir os objetivos.

Esta forma de ensino é muito utilizada em treinamentos de capacitação para utilização de equipamentos em uma fábrica, por exemplo. A ênfase maior desta forma é a programação das atividades em uma seqüência necessária para que os objetivos sejam alcançados.

4.2.3.3 Ensino programado

Para Falcão (1996) o ensino programado consiste num corpo de conceitos que sintetizam as idéias básicas sobre a apresentação de estímulos para a aprendizagem, condensando o planejamento, implementação e avaliação do ensino.

Esta forma parte do princípio que o conteúdo a ser ensinado aos alunos deve ser dividido em pequenos passos, interrelacionados que conduzirão ao objetivo final proposto. Nesta forma de ensino é possível reforçar a cada passo concluído o comportamento desejado. Com o ensino programado é possível gerar alto nível de aprendizagem por parte do aluno sem recorrer a contigências aversivas, situações em que o aluno rejeita a técnica de ensino

adotada em determinado assunto.

### 4.2.4 Avaliação

Na abordagem comportamentalista a avaliação está ligada diretamente aos objetivos estabelecidos. Ao final do programa ou da fase estabelecida o aluno será avaliado se atingiu os objetivos propostos. É possível, também, realizar avaliações intermediárias, onde são verificados se os comportamentos esperados estão sendo modelados e resultarão na consecução dos objetivos finais propostos.

A avaliação visa verificar se o comportamento desejado foi adquirido. A forma com que a avaliação será conduzida dependerá da situação planejada pelo professor. Se o comportamento esperado foi alcançado ele poderá avançar no ensino, caso o aluno não tenha apresentado o comportamento desejado o professor deverá retornar, de acordo com seu planejamento, ao ensino do objetivo que não foi alcançado.

## 4.3 Inatismo

A abordagem inatista ou preformista, também citada como humanista por Mizukami (1986) fundamenta-se que, ao nascer, a criança vem dotada das capacidades, aptidões e possibilidades que irão amadurecendo até a sua transformação em um adulto (Barros, 1996). Os inatistas partem do pressuposto que todos os eventos que ocorrem após o nascimento não são

relevantes e não interferem no desenvolvimento do ser humano.

Em comparação a corrente comportamentalista, a humanista difere por considerar as características internas do ser humano e relevar o ambiente ao plano inferior. A visão pessimista dada ao ser humano pela concepção ambientalista, por encará-lo como um ser passivo, sem nenhuma possibilidade de escolha é oposta a visão inatista que aceita as estruturas internas do ser humano, e coloca-o em uma situação ativa, que permite modificar a realidade de acordo com seus interesses.

Para Davis (1994) as qualidades e capacidades básicas individuais como a personalidade, valores, hábitos, crenças, conduta social já se encontrariam basicamente prontas, sofrendo poucas modificações ao longo da existência do ser humano. Nesta concepção o ambiente não interfere ou interfere o mínimo possível no ser humano. O desenvolvimento é espontâneo.

Como todas as características necessárias para que o aluno se desenvolva já são, nesta abordagem, inatas, muitos psicólogos desenvolveram o estudo das diferenças individuais e dos testes de inteligência e aptidões (Barros, 1996). Estas diferenças para Barros (1996) e para Davis (1994) criaram preconceitos prejudiciais ao trabalho em sala de aula.

Nesta concepção o papel do ambiente é bastante limitado. A educação é um

processo de dentro para fora. O aluno é o centro do processo e é ativo. Um representante importante desta corrente é Carl R. Rogers (1902 – 1987), psicólogo norte-americano, criador da "terapia centrada no cliente", que desenvolveu técnicas de aconselhamento que visava despertar as forças positivas de crescimento existentes dentro de todos os seres humanos (Barros, 1998).

Em sua obra Liberdade para aprender, publicada em 1969, Rogers apresentou suas idéias para o ensino. Para ele o aluno é um indivíduo que nasce com grandes possibilidades de desenvolvimento. Cabe ao professor acreditar nestas possibilidades e proporcionar um clima de liberdade favorável para que o aluno possa se desenvolver. Barros (1998) ressalta a qualidade da interação entre o professor e o aluno, pois quanto melhor for esta mais condições para desenvolvimento o aluno terá. Mizukami (1986) enfatiza esta visão defendida por Barros (1998) reforçando o professor como um facilitador da aprendizagem, um gerente que administra o clima e as condições que proporcionarão o desenvolvimento do aluno.

Esta ênfase dada por Rogers nas relações interpessoais conduz o educador a criar condições que promovam a aprendizagem. A qualidade da interação humana é muito mais significativa que o próprio ensinar. Rogers apresenta três condições fundamentais à aprendizagem, chamada de tríade rogeriana (Barros, 1998):

- ter empatia;
- aceitar incondicionalmente o aluno;
- ser autêntico.

Através da empatia o professor compreende melhor os sentimentos, desejos e ansiedades do aluno e permite que utilize as formas de comunicação mais adequada para que o aluno compreenda o que está sendo transmitido.

A aceitação incondicional do aluno consiste em aceitar o aluno como ele é. É respeitar, antes de tudo, as limitações e capacidades, sem oferecer ou esperar nada em troca. Esta aceitação incondicional e sem reservas faz com que o aluno acredite em suas potencialidades e sinta um clima favorável para desenvolver-se.

A autenticidade faz do professor uma pessoa verdadeira, sincera e honesta. A frente de seus alunos, dentro desta concepção, permite aumentar a confiança depositada nele. Torna o ambiente seguro e produtivo e faz o nível de ansiedade baixar (Barros, 1998).

Estas idéias rogerianas sobre a educação centrada no estudante é também conhecida como filosofia democrática. A responsabilidade de aprendizagem de novos conhecimentos é do próprio aluno. O objetivo da educação democrática, segundo Barros (1998) e Mizukami (1986) consiste em dar assistência aos

alunos para que se tornem pessoas independentes, responsáveis, autodeterminadas, capazes de discernir, aptas a aprender e solucionar seus problemas.

A corrente inatista valoriza o aluno, centro do processo de aprendizagem. Para Mizukami (1986) o conhecimento é inerente à atividade humana. O ser humano tem curiosidade natural para o conhecimento. O aluno como responsável pelo seu autodesenvolvimento criará sua própria experiência, que terá significados reais e concretos, ao mesmo tempo que perceberá que tudo é inacabado e que o conhecimento possui uma característica dinâmica.

### 4.3.1 Aprendizagem

Tomando as idéias rogerianas como modelo para esta corrente, a aprendizagem implica em desenvolver técnicas de conduzir os alunos sem, no entanto, estabelecer um caminho para que isto aconteça.

Segundo Mizukami (1986) a não-diretividade consiste num conjunto de técnicas que implementa a atitude básica de confiança e respeito pelo aluno. Para Puente (apud Mizukami, 1986, p.49)

> " a não-diretividade pretende ser um método não estruturante do processo de aprendizagem, pelo qual o professor se abstém de intervir diretamente no campo cognitivo e afetivo do aluno, introduzindo valores, objetivos etc., constituindo-se apenas num método informante do processo de aprendizagem do aluno, pelo qual o professor não dirige propriamente esse processo, mas apenas se limita a facilitar a comunicação do estudante consigo mesmo, para ele mesmo estruturar seu comportamento experimental. "

As condições que proporcionam a aprendizagem, nesta concepção, pode ser classificadas, segundo Flanders (apud Barros, 1998, p.78), em "influência direta e indireta do professor".

A influência direta consiste quando o professor disserta sobre determinado assunto, dá ordens, critica os alunos e justifica sua autoridade. Este tipo de influência é quando o professor, mesmo utilizando as idéias desta corrente, demonstra uma atitude comportamentalista que direciona a aprendizagem. A influência indireta dá-se quando o professor faz perguntas, aceita e utiliza as sugestões propostas pelos alunos, elogiando-os, encorajando-os e aceitando seus sentimentos. A influência indireta estimula a participação do aluno considerando suas respostas, sugestões, críticas e principalmente seus sentimentos. A qualidade da influência indireta está diretamente relacionada com o clima estabelecido pelo professor, e que proporciona aos alunos espaço para demonstrar seu aprendizado.

### 4.3.2 Relacionamento Professor-Aluno

Segundo Mizukami (1986) o professor não ensina: apenas cria condições para que os alunos aprendam, ele assume papel de facilitador da aprendizagem. Para Barros (1998) o professor limita-se a facilitar o auto-conhecimento do aluno, para que ele opte pelo seu caminho. O professor adota uma atitude centrada no aluno e esta deverá satisfazê-lo.

O professor não precisa, necessariamente, obter as competências e conhecimentos, pois estes serão desenvolvidos naturalmente, seguindo as fundamentações da abordagem inatista. Para Cória-Sabini (1986) a perspectiva inatista exige do professor a crença do pressuposto "pode-se confiar no aluno". Deste contexto o professor pode iniciar uma aula com uma pergunta ou com alguma curiosidade que os alunos possuam.

Não deve haver formalidade na aula elaborada. Ela irá se desenvolvendo com base nas opiniões e conhecimentos externizados, o professor apenas conduz, sem dirigir os alunos. A autodisciplina substitui a disciplina imposta externamente. Segundo Mizukami (1986) e Davis (1994) o aluno deve responsabilizar-se pelos objetivos referentes à aprendizagem, que têm significado para ele, e que, portanto, são os mais importantes.

Mahoney (apud Mizukami, 1986, p.49) organizou alguns princípios básicos desta abordagem, com base nas idéias rogerianas, voltadas para o centro do processo de aprendizagem: o aluno:

- "Todo aluno tem potencialidade para aprender e capacidade de desenvolver sua auto valorização;
- Todo aluno manifesta resistência à aprendizagem significativa;
- Se é pequena a resistência do aluno à aprendizagem significativa, então ele aumenta sua potencialidade para aprender;
- O aluno, ao realizar sua potencialidade para aprender, torna-se aberto à experiência, e reciprocamente;
- a auto-avaliação é função da capacidade organísmica de valoração;
- a criatividade é função da auto-avaliação;
- a autoconfiança é função da auto-avaliação;
- a independência é função da auto-avaliação".

### 4.3.3 Metodologia

Como o centro do processo é o aluno, as estratégias instrucionais assumem importância secundária. Cada educador, segundo Mizukami (1986), deve desenvolver um estilo próprio que facilite a aprendizagem dos alunos.

Como a característica básica dessa corrente é a ênfase atribuída a relação interpessoal estabelecida pelo professor com seus alunos, o clima favorável ao desenvolvimento é o suporte necessário para que os alunos encontrem ambiente para criticar, aperfeiçoar e, principalmente, aprender. Os objetivos educacionais não são tratados em seus aspectos formais.

Rogers (apud Mizukami, 1986, p. 54) propõe uma situação que:

- "restaure, estimule e intensifique a curiosidade do aluno;
- encoraje o aluno a escolher seus próprios interesses;
- promova todos os tipos de recursos;
- permita ao aluno fazer escolhas responsáveis quanto às suas próprias orientações, assim como assumir a responsabilidade das conseqüências de suas opções erradas, tanto quanto das certas;
- dê ao aluno papel participante na formação e na construção de todo o programa de que ele é parte;
- promova interação entre meios reais;
- focalize, por meio de tal interação, problemas reais;
- desenvolva o aluno autodisciplinado e crítico, capaz de avaliar tanto as suas quanto as contribuições dos outros;
- capacite o aluno a adaptar-se inteligente, flexível e criativamente a novas situações problemáticas do futuro".

### 4.3.4 Avaliação

Dentro da concepção inatista e segundo as idéias rogerianas não há nenhuma padronização que permita avaliar os alunos ou as competências do professor. É estimulada a auto-avaliação pois só o próprio indivíduo pode conhecer sua

experiência e, em conseqüência, julgar de acordo com seus critérios internos se a aprendizagem está ou não acontecendo.

Para Rogers (apud Mizukami, 1986, p. 55)

> "quando reunimos em um esquema elementos tais como currículo preestabelecido, "deveres idênticos" para todos os alunos, preleções como quase único modo de instrução, testes padronizados pelos quais são avaliados externamente todos os estudantes, e notas dadas pelo professor, como modo de medir a aprendizagem, então, quase podemos garantir que a aprendizagem dotada de significação será reduzida à sua expressão mais simples".

Nesta corrente o aluno assume a responsabilidade pelas formas de controle de sua aprendizagem, definindo e aplicando os critérios necessários para julgar se a aprendizagem está ou não sendo satisfatória. Se não estiver deverá buscar novos conhecimentos utilizando seus critérios internos de auto-avaliação, independência e autoconfiança.

### 4.4 Interacionismo

A concepção interacionista ou cognitivista não valoriza nem o meio que o aluno se encontra, como o empirismo, nem somente o aluno, desconsiderando o meio, como a inatista, mas considera que o aluno interfere no meio, tanto quanto o meio interfere no aluno. Nesta abordagem o desenvolvimento do aluno é uma interação do meio ambiente onde ele se encontra com os fatores subjetivos internos que possui.

Para Mizukami (1986) uma abordagem cognitivista implica, dentre outros aspectos, estudar cientificamente a aprendizagem como sendo mais que um

produto do ambiente, das pessoas ou de fatores que são externos ao aluno.

Esta concepção considera como o aluno trabalha com os estímulos ambientais, como interage com situações sociais, como resolve problemas, como organiza os dados e como adquire e emprega os símbolos verbais. A abordagem interacionista se preocupa como o aluno capta, processa e integra as informações nas relações sociais. Segundo Davis (1994) as experiências anteriores também servem para novas construções, sempre relacionando o aluno com o ambiente.

Como representantes mais significativos desta abordagem têm-se Jean Piaget (1896 – 1980), que fundamentou suas pesquisas procurando explicar as fases que representam o desenvolvimento da criança e sua teoria interacionista é conhecida como construtivismo, e Lev Semenoviich Vygotsky (1896 – 1934), que pesquisou a interação as estruturas internas da criança e as condições sociais em que ela vive. Para Vygotsky o contato com os membros mais experientes de seu grupo social faz com que a criança se aproprie do conhecimento existente e disponível. A esta teoria foi dado o nome de sócio-interacionismo.

### 4.4.1 Construtivismo

Segundo Matui (1995) o construtivismo fundamenta a construção da mente e do conhecimento sobre as bases anteriores, num processo extremamente

dinâmico e reversível de equilibração majorante. Para o construtivismo a criança age espontaneamente sobre o meio e possui uma lógica de funcionamento mental que difere, qualitativamente, da lógica de funcionamento mental de um adulto. Este modo de funcionamento intelectual próprio não a impede de adaptar-se e organizar, de sua maneira, suas experiências.

O alicerce do construtivismo, na concepção de Piaget, é o equilíbrio. A criança sempre procura manter o estado de equilíbrio diante das perturbações que o meio lhe causa. Segundo Davis (1994) o processo dinâmico e constante do organismo buscar um novo e superior estado de equilíbrio é denominado de equilibração majorante.

Na teoria de Piaget o desenvolvimento cognitivo é um processo constante de desequilíbrios e equilíbrios. Diante de uma situação nova, desconhecida o aluno sofre uma mudança no seu estado repouso, há um desequilíbrio entre o organismo (estruturas internas) e o meio. Para alcançar, novamente, o estado de equilíbrio dois mecanismos são acionados: a assimilação e a acomodação.

O primeiro mecanismo se refere ao desenvolvimento de ações destinadas a atribuir significações, com base nas experiências anteriores, aos elementos do meio com os quais a criança interage. Este mecanismo aproxima a nova situação a alguma já existente dentro do organismo da criança. Ocorre uma busca nas estruturas internas verificando se algo semelhante já não se

encontra incorporado (Matui, 1995). Um exemplo prático deste mecanismo é quando a criança já sabe o que é um cachorro e que tem quatro patas. Diante de um novo animal de quatro patas, uma vaca por exemplo, a criança poderá nomear este novo animal também de cachorro, pois ambos tem algo semelhante: possuem quatro patas.

O segundo mecanismo, a acomodação, diz respeito ao restabelecimento de um equilíbrio superior com o meio ambiente. Para Matui (1995) está relacionada a mudança, não do objeto, mas da própria pessoa. É quando compreendemos determinadas situações e a partir deste momento elas passam a fazer parte das nossas estruturas internas. É quando o conhecimento é sedimentado (Davis, 1994).

Para Piaget o desenvolvimento cognitivo é contínuo e evolui obedecendo estágios pré-determinados e sucessivos de assimilação e acomodação que proporcionam a criança o amadurecimento de suas estruturas internas. Piaget definiu as seguintes etapas: sensoriomotora (0 até 2 anos), pré-operatória (2 até 7 anos), operatório-concreta (7 até 11 anos) e operatório-formal (11 em diante). Estes limites estabelecidos, na verdade refletem a idade mental da criança e representam a média de um grupo. Pode haver variações em decorrência do ambiente em que as crianças vivem.

Na etapa sensoriomotora a criança utiliza esquemas motores para resolver

seus problemas. Segundo Davis (1994) apesar da criança possuir uma conduta inteligente, considera-se que ela ainda não possui pensamento e não dispõe de capacidade de representar eventos, referir-se ao passado e ao futuro. Atua sobre os objetos de forma pré-lógica colocando um sobre o outro, um dentro do outro. Mesmo não tendo uma seqüência coerente, Davis (1994) enfatiza que esta atitude, nesta fase, já representa algo que futuramente já amadurecido, representará um esquema lógico.

Na fase pré-operatória a característica mais importante é o aparecimento da linguagem oral. Ela permite a criança utilizar os esquemas da fase anterior, a sensoriomotora, adicionado dos esquemas representativos ou simbólicos, ou seja, aqueles que representam uma idéia a respeito de algo. Utiliza a substituição de objetos, ações, situações e pessoas por símbolos. Tem origem o pensamento sustentado por conceitos.

A fase operatório-concreta é a que o pensamento lógico e objetivo se ressaltam. A criança consegue aplicar seu pensamento lógico em problemas concretos. O conhecimento que ela constrói é próximo e compatível ao mundo que a cerca. Ela é capaz de perceber quantidades e dissociá-las entre si, como volume, massa e peso. É uma fase onde ainda não consegue abstrair, ela necessita do objeto concreto para inferir novos conhecimentos.

Na última etapa, descrita por Piaget, a operatório-formal, as estruturas

cognitivas já amadurecidas permitem, a criança, aplicar o pensamento lógico em qualquer classe de problemas. Consegue abstrair e já não necessita do objeto concreto para inferir novos conhecimentos (Mizukami, 1986). A liberdade do pensamento em relação ao mundo concreto permite formular hipóteses.

Observa-se que a evolução do desenvolvimento cognitivo é na verdade um amadurecimento dos esquemas mentais. Cada fase é pré-requisito para outra. Não é possível deixar de passar por uma delas. O que pode acontecer, segundo Davis (1994), é a criança permanecer em uma etapa um período muito curto, menor que a média do seu grupo.

4.4.1.1 Aprendizagem

Dentro das idéias descritas por Piaget, para Mizukami (1986), o ensino deve priorizar as atividades do sujeito, considerando-o inserido numa situação social, e as ações que o meio exerce sobre este. Esta concepção de aprendizagem permite abertura de idéias e comporta possibilidades de novas indagações.

Segundo Silva (2000) a aprendizagem consiste em assimilar o objeto a esquemas mentais. O aluno aprende dependendo da esquematização presente, do estágio e da forma de relacionamento atual com o meio. Como conseqüência, o ensino deve assumir várias formas durante o seu desenvolvimento.

A aprendizagem é baseada no ensaio, no erro, na pesquisa, na investigação, na solução de problemas por parte do aluno. A ênfase maior é dada a aprendizagem por descoberta que permite ao estudante atingir um nível de compreensão que ultrapassa, em muito, a memorização de determinado assunto (Barros, 1998). A utilização de fórmulas, nomenclaturas, definições são eliminadas desta abordagem. A aprendizagem só se realiza quando o aluno elabora seu próprio conhecimento.

No construtivismo a aprendizagem é um organizador dos dados da experiência e deve evoluir progressivamente, auxiliando a fixação do conhecimento e evitando a formação de hábitos. Para Mizukami (1986) o ensino dos fatos deve ser substituído pelo ensino das relações, desta maneira desenvolve-se a inteligência.

4.4.1.2 Relacionamento Professor-Aluno

O professor deverá dar aos alunos oportunidade para que eles participem na elaboração dos objetivos que pretendem alcançar, no planejamento e na execução das atividades que irão desenvolver e na avaliação. Desta maneira haverá um envolvimento maior do aluno que na corrente empirista. Segundo Cória-Sabini (1986) o professor, ainda, deverá estimular a discussão, a livre troca de idéias, sem críticas e preconceitos.

O professor deverá criar situações que propiciem uma reciprocidade intelectual

e cooperação ao mesmo tempo moral e racional (Mizukami, 1986). Ele deve diversificar as atividades desenvolvidas evitando a rotina, fixação de respostas e hábitos. Ele deve propor problemas sem ensinar as soluções. Será um orientador para desenvolver, no aluno, a autonomia e autoconfiança.

O professor, para Mizukami (1986), deve assumir um papel de investigador, pesquisador, orientador, coordenador, levando o aluno a trabalhar de forma independente. A convivência com os alunos, a observação de comportamentos, a constante interrogação, permite ao professor uma aproximação e auxílio na aprendizagem e desenvolvimento do aluno.

O aluno deve ser ativo, independente, investigador e observador. Para Mizukami (1986) ele deve observar, experimentar, comparar, relacionar, analisar, justapor, compor, encaixar, levantar hipóteses, argumentar, entre outras atividades. Quando percebe uma nova situação ele deve, antes de tudo, buscar a resposta e não esperá-la pronta. Esta busca da solução faz com que ele construa o seu próprio conhecimento.

4.4.1.3 Metodologia

Não existe uma metodologia definida dentro do modelo piagetiano. O construtivismo definido por Piaget valoriza a inteligência e proporciona um grande desenvolvimento humano. Para Aebli (apud Mizukami, 1986, p.79) "uma didática científica deve ter por finalidade deduzir do conhecimento

psicológico dos processos de formação intelectual as técnicas metodológicas mais adequadas para produzir tais processos".

Toda estratégia de ensino empregada deve estar fundamentada na investigação e na pesquisa, base da teoria piagetiana. O problema será um auto-regulador das atividades desenvolvidas, pois cada situação presenciada durante o processo de busca da solução, possibilita ao aluno uma nova experiência e um novo conhecimento.

Piaget estimula os trabalhos em equipe, pois esta estratégia permite ao aluno ter a visão do grupo e ser este um elemento importante na socialização do indivíduo, base para a teoria desenvolvida por Vygotsky. Os membros do grupo funcionam como controle lógico do pensamento individual (Mizukami, 1986).

O material didático utilizado deve ser flexível e possibilitar combinações e realizações adaptáveis a cada fase do desenvolvimento. O aluno deve construir o seu próprio material. A utilização de recursos tecnológicos não são suficientes para desenvolver a atividade operatória, pois sempre figuram os conteúdos a serem transmitidos. Deve-se incentivar o contato com o objeto concreto, real e vivo.

4.4.1.4 Avaliação

Na abordagem empirista o aluno é avaliado a partir de um padrão, de uma

regra. Este tipo de avaliação faz com que repetidos insucessos possam ter conseqüências desastrosas na medida em que levam à perda da confiança por parte do aluno. Na concepção construtivista avaliações tradicionais como testes, provas, exames, entre outros não encontram fundamentação para serem aplicadas.

Piaget sugere extrair parâmetros estabelecidos pela teoria e verificar se o aluno já adquiriu noções, realizou operações, apresentou soluções adequadas e se conservou o conhecimento. Para Mizukami (1986) o rendimento poderá ser avaliado de acordo com a sua aproximação a uma norma qualitativa pretendida. O controle de aproveitamento deve ser múltiplo considerando a qualidade das soluções encontradas.

Deve ser considerada as respostas certas e, também, as erradas, incompletas, distorcidas, pois cada aluno possui uma ótica do problema que acabou de resolver. As respostas apresentadas devem ser consideradas de forma qualitativa e não quantitativa. Não deve haver pressão de desempenho escolar, pois a aprendizagem não é linear e idêntica para todos os alunos, pois cada um possui uma interpretação do mundo, dos fatos e da causalidade.

### 4.4.2 Sócio-Interacionismo

Para o sócio-interacionismo, defendido por Vygotsky, há uma contínua interação entre as estruturas internas da criança e as condições sociais em que

ela vive (Barros, 1996). O convívio com pessoas mais experientes faz com que a criança se aproprie do conhecimento disponível na sociedade. Para Vygotsky, o conhecimento é construído paulatinamente e é representado por instrumentos físicos, objetos concretos como por exemplo: mesa, casa, faca, entre outros, e instrumentos simbólicos, objetos abstratos, como por exemplo: cultura, valores, costumes, crenças, tradições, religião, conhecimentos, entre outros (Moreira, 1999).

Comparativamente ao construtivismo, defendido por Piaget, que privilegia a maturação biológica e postula que o desenvolvimento segue uma seqüência fixa e universal de estágios (Davis, 1994), o sócio-interacionismo não aceita a possibilidade de existir uma seqüência de estágios cognitivos. Para Moreira (1999) esta interação social é fundamental para o desenvolvimento cognitivo e lingüístico de qualquer indivíduo. Porém, seus mecanismos são difíceis de identificar, qualificar e quantificar com precisão.

O alicerce da teoria de Vygotsky são os chamados processos mentais elementares e superiores. Os processos mentais elementares se referem àqueles do estágio sensório-motor da teoria de Piaget. Para Silva (2000) são os processos derivados do capital genético da espécie, da maturação biológica e da experiência da criança com seu ambiente físico. Os processos mentais superiores são construídos ao longo da vida social do homem e possibilitam a capacidade de solucionar problemas, de armazenar e utilizar adequadamente a

memória e permitem a formação de novos conceitos a partir das relações sociais.

Para Vygotsky diante de uma relação social o indivíduo passa por duas etapas: a interiorização e a transformação. A interiorização é como o indivíduo reconstrói internamente um processo externo, ou seja, é a apropriação de algo disponível na sociedade. A transformação é a reorganização dos processos mentais. Pode-se fazer um paralelo as etapas de assimilação e acomodação propostas por Piaget no construtivismo. A interiorização e a assimilação se referem ao modelo, conhecimento de algo novo, que necessita ser anexado ao restante já existente. A transformação e a acomodação estão relacionadas a fixação, sedimentação do conhecimento, reorganizando aquele já existente dentro do indivíduo (Moreira, 1999).

A conversão das relações sociais em funções psicológicas se dá através da mediação com a utilização de instrumentos e signos. Segundo Oliveira (1997) o instrumento é um elemento interposto entre o trabalhador e objeto de seu trabalho, ampliando as possibilidades de transformação da natureza. O instrumento é confeccionado de acordo com o objetivo que se deseja atingir. Os signos são elementos que representam ou expressam outros objetos, eventos ou situações. Para Moreira (1999) existem três tipos de signos: os indicadores, que apresentam uma relação de causa e efeito com aquilo que significam, por exemplo fumaça indica fogo, porque é causada pelo fogo; os

icônicos, são imagens ou desenhos daquilo que significam, como por exemplo a figura de uma casa, de um carro ou de um avião e os simbólicos, que têm uma relação abstrata com o que representam, por exemplo as palavras, são signos lingüísticos, os números, são signos matemáticos.

Para Vygotsky, desenvolvimento e aprendizagem são fenômenos distintos e interdependentes (Davis, 1994). Vygotsky considera três teorias para relacionar desenvolvimento e aprendizagem. A primeira, defendida por Piaget, o desenvolvimento é visto como um processo maturacional que ocorre antes da aprendizagem. Na segunda, o desenvolvimento e a aprendizagem são vistos como na teoria comportamentalista, ou seja, um acúmulo de respostas aprendidas. Na terceira, desenvolvimento e aprendizagem são vistos como processos independentes que se interagem mutuamente.

Neste processo de desenvolvimento e aprendizagem Vygotsky (apud Moreira, 1999, p. 116) define a zona de desenvolvimento proximal (ZDP) como

> "a distância entre o nível de desenvolvimento cognitivo real do indivíduo, tal como medido por sua capacidade de resolver problemas independentemente, e o seu nível de desenvolvimento potencial, tal como medido através da solução de problemas sob orientação ou em colaboração com companheiros mais capazes".

Este conceito de ZDP é fundamental para um ensino efetivo. Através da ZDP é possível planejar as situações de ensino e avaliar os progressos individuais. Na teoria de Vygotsky o desenvolvimento é a apropriação ativa do conhecimento disponível na sociedade em que a criança nasceu. Através da experiência ela

aprende a interagir com a sociedade ao mesmo tempo que se posiciona a frente como um ser crítico. Com a mediação, instrumentos e signos o funcionamento dos processos mentais superiores são refinados, permitindo ao ser processar as informações de uma forma muito mais elaborada.

#### 4.4.2.1 Aprendizagem

Para Vygotsky a aprendizagem é necessária para que haja o desenvolvimento cognitivo do ser. Segundo Driscoll (apud Moreira, 1999, p. 119)

> "a interação social que provoca aprendizagem deve ocorrer dentro da zona de desenvolvimento proximal, mas, ao mesmo tempo, tem um papel importante na determinação dos limites dessa zona. O limite inferior é, por definição, fixado pelo nível real de desenvolvimento do aprendiz. O superior é determinado por processos instrucionais que podem ocorrer no brincar, no ensino formal ou informal, no trabalho. Independente do contexto, o importante é a interação social".

A aprendizagem deve ser orientada para níveis de desenvolvimento ainda não alcançados, mas que estejam dentro da ZDP. A linguagem, o sistema mais importante de signos para o desenvolvimento, deve ser utilizada e bastante estimulada. A interação social que caracterizará o ensino deve ser compartilhada com os materiais educativos previstos em currículo. O ensino deve estar a frente do desenvolvimento cognitivo, e a aprendizagem deve estar avançada em relação a este desenvolvimento.

No construtivismo a aprendizagem procede do individual para o social. No sócio-interacionismo a aprendizagem dá-se do social para o individual. Para Moreira (1999) o ensino se consuma quando aluno e professor compartilham

os mesmos significados integrantes da mesma relação social.

4.4.2.2 Relacionamento Professor-Aluno

O professor é visto, por Vygotsky, como uma pessoa mais experiente que teve um maior contato com a sociedade e, portanto, já conviveu com mais situações sociais que o aluno. Seu papel fundamental é de mediador na aquisição de significados contextualmente aceitos, pois o professor já possui estes significados internalizados. Para Silva (2000) o professor deve ser capaz de ajudar o aluno a entender um determinado assunto e, ao mesmo tempo, relacioná-lo ao conteúdo com experiências pessoais e o contexto no qual o conhecimento será aplicado. Deve atuar dentro da ZDP identificando o potencial do aluno que ainda não foi desenvolvido.

O professor deve intervir no desenvolvimento do aluno, questionando respostas, observando o comportamento apresentado diante de determinadas relações sociais e a interferência de outros alunos no desenvolvimento. Deve verificar a evolução da experiência obtida por seus alunos através do compartilhamento dos conhecimentos adquiridos.

Ao aluno cabe construir a compreensão do assunto e verificar se os significados captados são aqueles transmitidos pelo professor. O aluno deve interagir com o meio de maneira que sempre acrescente e forneça conhecimentos.

4.4.2.3 Metodologia

Como no construtivismo, aqui, também, não há uma metodologia específica, pronta para ser aplicada. Cabe ao professor considerar em cada situação de ensino as estratégias que mais favorecem ao aprendizado. Com base na teoria de Vygotsky, a estratégia de ensino que for baseada na interação social, com intercâmbio de significados e que esteja dentro da zona de desenvolvimento proximal do aluno será uma estratégia eficaz.

O professor deve proporcionar interação entre todos os alunos. A observação constante evita o isolamento, prejudicial a aprendizagem, e permite ao professor guiar seus alunos. Todos devem falar e ter oportunidade para falar (Moreira, 1999).

Nesta concepção o professor é visto como um ser mais experiente que já internalizou significados socialmente aceitos. Ele deve fazer uso desta experiência para elaborar situações sociais que propiciem, ao aluno, ambiente favorável a interação social e humana.

4.4.2.4 Avaliação

Nesta teoria o processo de avaliação consiste na auto-avaliação ou avaliação mútua (Silva, 2000). O professor, em constante observação, pode acompanhar o desenvolvimento de seus alunos e verificar se os significados transmitidos

foram captados dentro do contexto da área de conhecimento em questão (Moreira, 1999).

Como na concepção construtivista, avaliações tradicionais como testes, provas, exames, entre outros não encontram fundamentação para serem aplicadas, pois somente o professor como o aluno saberão de suas dificuldades e da quantidade e qualidade dos conhecimentos transmitidos em uma interação social.

Todo o processo de avaliação não deve ser desenvolvido de maneira inflexível, que não possibilite ao professor considerar os fatores da interação social. A avaliação deve ser contínua e oportuna (Oliveira, 1997). O maior peso deve ser considerado sobre o compartilhamento de significados, pois para Vygotsky, esta é a essência de toda a sua pesquisa.

### 4.5 Considerações Finais

As abordagens apresentadas neste capítulo representam as principais teorias de aprendizagem em utilização. Cada uma com suas características, vantagens e desvantagens e, relacionando ou não, o aluno e o meio em que ele se encontra.

Cada teoria é fundamentada em uma metodologia, forma como o conhecimento é transmitido, em uma forma de aprendizagem, maneira como o

aluno recebe este conhecimento, estabelece, também, determinadas tarefas ao professor, pois este é o elo de ligação entre o saber e o aluno.

Mesmo com as divergências e convergências apresentadas em cada teoria, a figura do professor é a principal. É ele que, de uma forma ou outra, permitirá ao aluno a aquisição de habilidades, conhecimentos, cultura, entre outros. O aluno, que poderá ser ativo ou passivo, é sempre o objetivo principal de cada uma das abordagens descritas.

# 5 MODELO PROPOSTO

O objetivo deste capítulo é descrever o modelo de um STI para a disciplina de programação estruturada interrelacionando-o com a revisão bibliográfica apresentada nos capítulos anteriores e com os exemplos de STI citados no capítulo 3. As telas do protótipo construído a partir do modelo proposto seguem as explicações ilustrando, melhor, o funcionamento do STI.

## 5.1 Fundamentos do Modelo Proposto

Os STI que auxiliam no ensino de programação para iniciantes têm se ocupado, ao longo dos anos, em proporcionar instrução aos alunos de como desenvolver programas e assisti-los para aprenderem habilidades de programação de um modo geral (Giraffa, 1996). A principal característica destes sistemas é fazer com que o estudante desenvolva um programa de forma independente, ou seja, há um enunciado e o aluno, através de um editor de texto, como por exemplo o bloco de notas do Microsoft Windows®, desenvolve o algoritmo que resolverá o problema proposto no enunciado.

Esses STI, segundo Chang *et al* (1996) possuem a característica de identificar os erros, sintáticos e semânticos, nas estrutura básicas de programação e mostrá-los ao aluno, especificando onde e porque estes erros aconteceram, e depois propor correções. Esses STI, nem sempre, interrelacionam o erro identificado à teoria do assunto e a um exemplo correto que utilize a mesma

estrutura de programação. Esta estratégia é conhecida como estratégia de generalização (Giraffa, 1996).

Esta estratégia mostra seus inconvenientes pois o aluno iniciante pode ter uma atitude depressiva e frustada ao começar a fazer seu algoritmo e se deparar com uma grande coleção de erros, muitas vezes encadeados, ou seja, um simples erro em uma linha de código ocasiona diversos erros nas linhas seguintes.

O STI que utiliza a estratégia de generalização possui outro inconveniente de que a solução apresentada pelo sistema, geralmente, se reduz a apenas uma, desconsiderando outras possibilidades de resolução do problema apresentada pelo aluno (Giraffa, 1996).

Para evitar estes problemas, ocasionados pelo uso da estratégia de generalização, os STI voltados ao ensino de programação utilizam a estratégia de acabamento proposta por Van Merrienboer (apud Chang *et al*, 1996, p. 58).

Nesta estratégia um programa incompleto, porém muito bem elaborado, é apresentado ao aluno para ser finalizado. As modificações realizadas não devem tirar a funcionalidade do programa. Esta estratégia é baseada na pesquisa realizada por Deimel & Moffat (apud Chang *et al*, 1996, p. 91) onde devem ser obedecidos quatro passos introdutórios em uma aula de

programação:

1. "Fazer com que o aluno execute um programa e avalie seus aspectos positivos e suas limitações;
2. Fazer com que o aluno estude estruturas de programas precisas;
3. Fazer com que o aluno seja capaz de modificar e entender o programa recebido;
4. Fazer o aluno escrever um novo programa independente de estrutura prévia".

Estes passos propostos por Deimel & Moffat modelam o que na prática alguns professores de programação já os fazem. Depois da explicação teórica sobre o assunto e a apresentação de um exemplo, é dado um enunciado seguido de um exercício, incompleto, que necessita da complementação para que se torne um algoritmo completo e preciso e que resolva o problema proposto no enunciado.

Segundo Giraffa (1998) em uma sessão de ensino de programação de computadores, tradicional ou com o auxílio de computador, após terminar a discussão em cima de um exemplo apresentado, o professor simplesmente escolhe outro exemplo e continua a sessão. O grande problema é como selecionar adequadamente o próximo exemplo. Em uma turma real, o professor usa a sua experiência. Nos sistemas tutores, esta escolha ainda não está formalizada, pois os próprios professores não sabem dizer exatamente quais os critérios usados para tal.

Woolf (1988) classifica os STI em fortes e fracos. Um STI fraco deve ser capaz de ministrar o conteúdo, propor questões a serem resolvidas pelo aluno e

reconhecer e corrigir erros. Um STI forte deve, além disto, ser capaz de resolver as questões propostas aos alunos e utilizar o conteúdo na resolução dos problemas, identificando na teoria o erro apresentado, reforçando desta maneira o aprendizado.

## 5.2 Caracterização do Modelo

Com base nas considerações iniciais apresentadas e na revisão bibliográfica feita nos capítulos anteriores o modelo do STI para ser aplicado ao ensino de programação estruturada será um STI, segundo Woolf (1988) forte, pois será capaz de resolver as questões propostas aos alunos e utilizará o conteúdo da disciplina na identificação do erro apresentado durante a construção dos algoritmos.

A estratégia de ensino adotada será, segundo Chang *et al* (1996) a estratégia do acabamento, nos exercícios classificados como fácil e médio na fase inicial de desenvolvimento, e a estratégia da generalização, segundo Giraffa (1998), nos exercícios classificados como difícil e muito difícil na fase de aprimoramento.

A utilização destas duas estratégias combinadas reforça o que Direne & Pimentel (1998) levantou a respeito da alta carga cognitiva que a programação representa aos iniciantes. A falta de perícia identificada por estes autores será minimizada com a aplicação da estratégia de acabamento, onde o aluno

completará um algoritmo com os passos faltantes para que este resolva o problema proposto e adquirirá conhecimentos necessários para que prossiga no desenvolvimento de algoritmos.

A estratégia da generalização será aplicada no estágio onde o aluno já apresenta condições e domínio pleno das estruturas básicas de programação. Nesta fase o aluno encontrará exercícios classificados como difícil e muito difícil, de maneira que a perícia adquirida no estágio anterior lhe proporcionará conhecimentos adequados no desenvolvimento dos exercícios. Durante as duas fases haverá, no modelo do estudante, um dispositivo, que identificando um erro na construção do algoritmo remeterá o aluno ao tópico abordado na disciplina, buscando no modelo do especialista a teoria relacionada ao erro cometido e apresentando exemplos adequados ao nível de desenvolvimento em que se encontra o aprendiz.

Segundo a estrutura básica de um STI proposta por Kaplan & Rock (1995) e Giraffa (1996), onde os STI possuem quatro modelos ou módulos, a saber: Modelo do Especialista; Modelo do Estudante; Modelo Pedagógico e o Modelo de Interface. O modelo proposto de STI estará estruturado com base nestes módulos da seguinte forma:

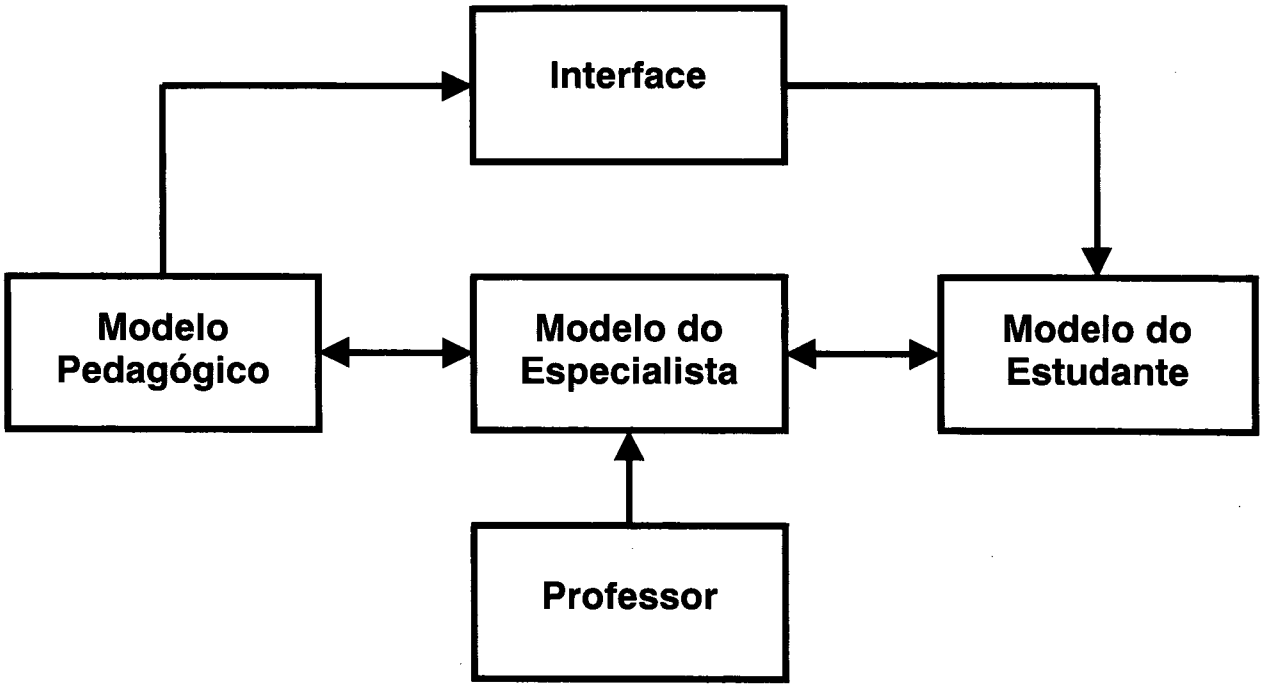



**Figura 4** - Diagrama em blocos do Modelo Proposto de STI.

Com base no diagrama acima exposto será detalhado nos itens a seguir a função de cada bloco, especificando suas principais atividades, objetivos e características.

### 5.3 Estrutura e Funcionamento do Modelo

A estrutura do modelo proposto é baseada no objetivo de possibilitar a automação da escolha do próximo algoritmo a ser trabalhado pelo aluno, simulando a experiência do professor, a fim de minimizar o tempo de aprendizado, utilizando a estratégia do acabamento na fase inicial e a de generalização, na fase de aprimoramento.

### 5.3.1 Professor

Neste bloco está representada a figura do professor responsável pela alimentação dos conhecimentos que ficarão armazenados dentro do sistema. O professor possui, neste modelo, a função de inserir novos conhecimentos e manter atualizado os conhecimentos anteriormente cadastrados. A inclusão do professor como parte do STI denota a figura central representada pelo profissional responsável pela disciplina.

No protótipo desenvolvido a comunicação do professor com o modelo do especialista dá-se através da seguinte tela:

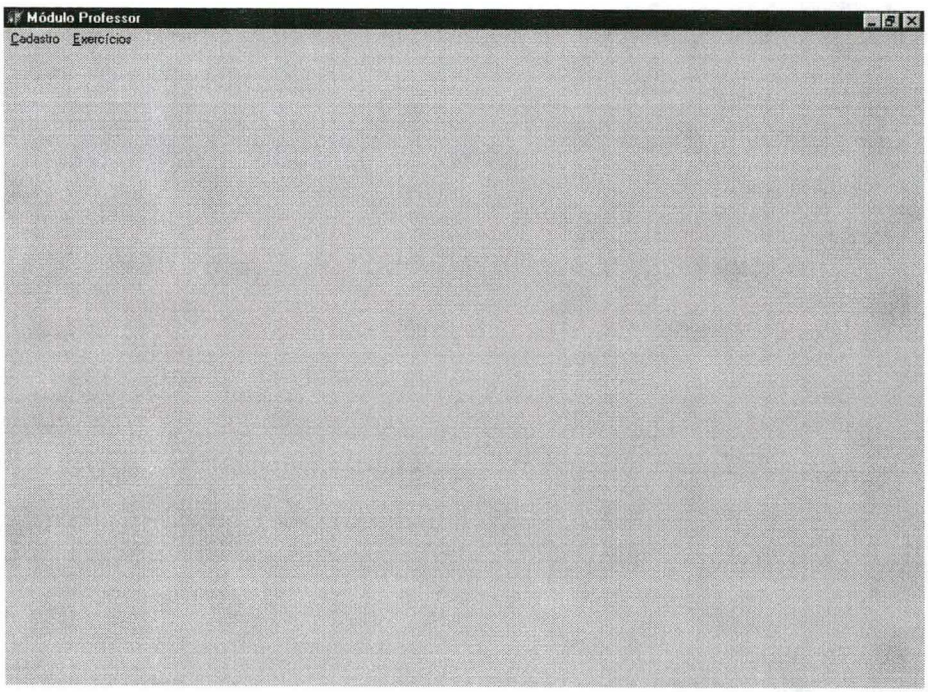



**Figura 5** - Tela do Professor.

Nesta tela pode-se na função cadastro, cadastrar professor, turma e disciplina, além da função sair, que finaliza o aplicativo. Na opção exercícios, pode-se cadastrar os conhecimentos relativos as teorias e os exercícios vinculados a esta. A figura 6 ilustra as opções existentes em cadastro e exercícios.

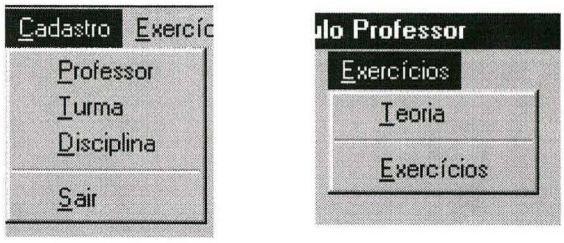



**Figura 6** - Opções do Menu do Professor.

No cadastro de Professor, além do nome do professor é importante a colocação da senha, que permitirá, somente o professor manter a base de conhecimentos. A figura 7 mostra a tela de cadastro de professor.

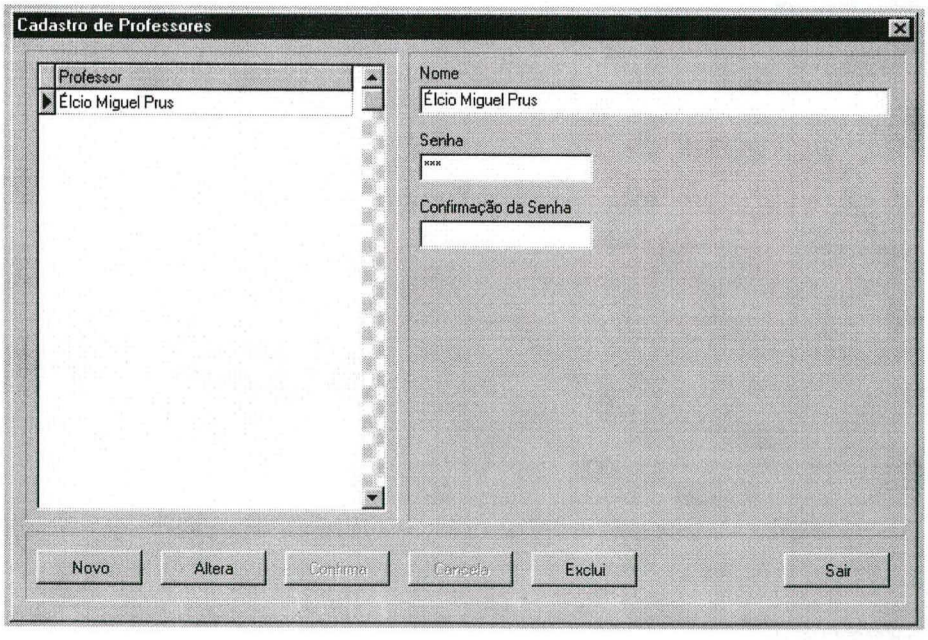



**Figura 7** – Tela de Cadastro de Professor.

O cadastro de turma, ilustrado pela figura 8, pode-se colocar o nome ou abreviatura pela qual a turma é representada. No campo descrição é utilizado para complementar com informações relevantes da turma cadastrada, como por exemplo o número de alunos, a média atingida na última avaliação, ou qualquer outro dado de relevância da turma.

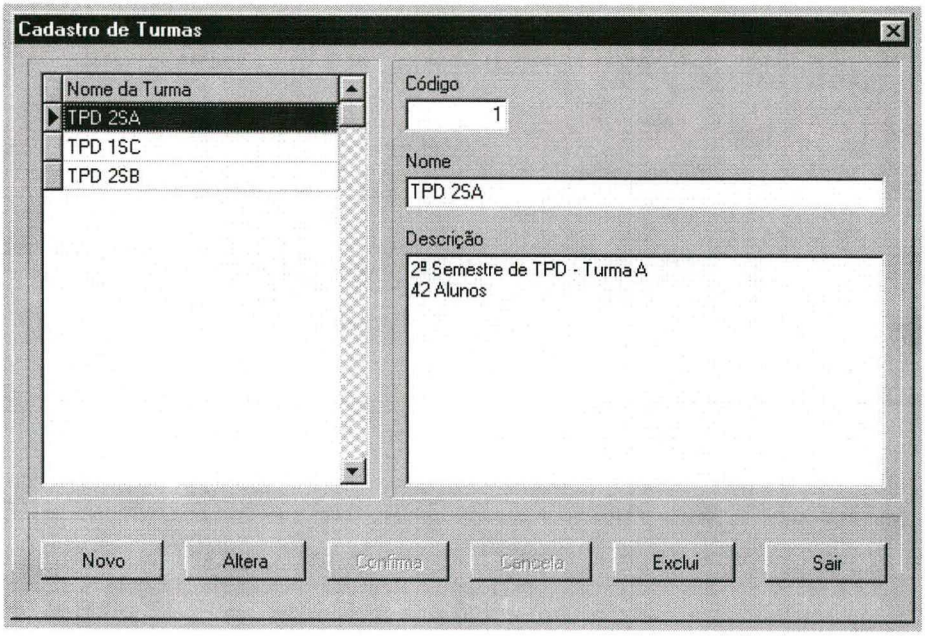



**Figura 8** – Tela de Cadastro de Turma.

No cadastro de disciplina, ilustrado pela figura 9, permite, futuramente, que o protótipo apresentado venha a ser utilizado em outras disciplinas. Inicialmente, como definido nos objetivos desta pesquisa, está restrito a disciplina de Algoritmos e Programação I.

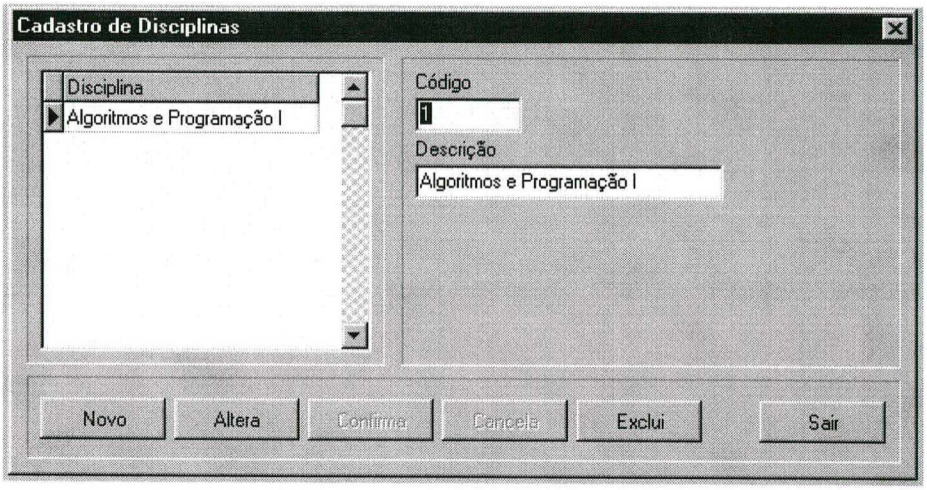



**Figura 9** – Tela de Cadastro de Disciplina.

O módulo professor no modelo proposto permite personalizar o sistema de acordo com algumas características básicas de identificação do professor, da disciplina e da turma que acessará o STI. A flexibilização de disciplina e de turma, permitindo a inclusão de mais de uma de ambas, já aponta para a possibilidade deste modelo ser aproveitado em outros conteúdos, senão aquele descrito nos objetivos desta pesquisa.

### 5.3.2 O Modelo do Especialista

O conteúdo do modelo do especialista dividido em teoria e seus exercícios. O modelo entidade relacionamento ou o diagrama entidade relacionamento (DER), definido por Yourdon (1999) onde dois objetos podem se relacionar em quatro cardinalidades: um-para-um, um-para-muitos, muitos-para-um ou muitos-para-muitos, melhor se adapta para representar o relacionamento

existente entre a teoria e os exercícios apresentados. A figura 10 representa este relacionamento:

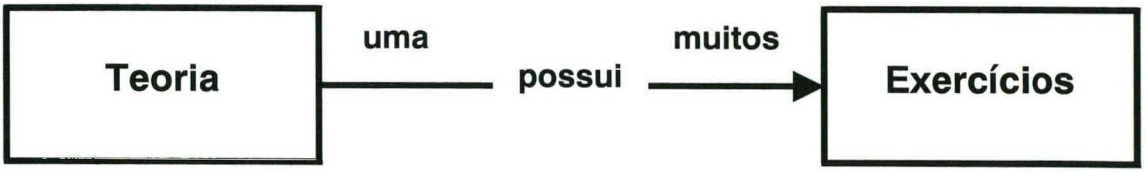



**Figura 10** – Relacionamento entre teoria e exercícios.

Da forma apresentada uma determinada teoria pode possuir diversos exercícios relacionados a ela. Neste modelo o conhecimento é armazenado utilizando as regras de produção, de acordo com Woolf (1988) e Giraffa (1998), pois o domínio a ser representado é orientado a uma tarefa, que é a resolução dos exercícios.

A teoria e os exercícios ficam armazenados em uma base de dados. O conhecimento do especialista é armazenado através dos valores (condicionantes) dados para que determinado exercício seja considerado correto ou não. Estes valores, linhas médias de código, tempo mínimo e máximo para execução do exercícios, serão referências utilizadas na correção da solução apresentada.

Estes critérios serão armazenados neste modelo na mesma base de dados que o exercício, porém, durante a correção do mesmo, alimentarão as regras de

produção de correção, responsáveis pela obtenção do índice de aproveitamento do aluno.

A figura 11 mostra a tela que permite o professor cadastrar a teoria.

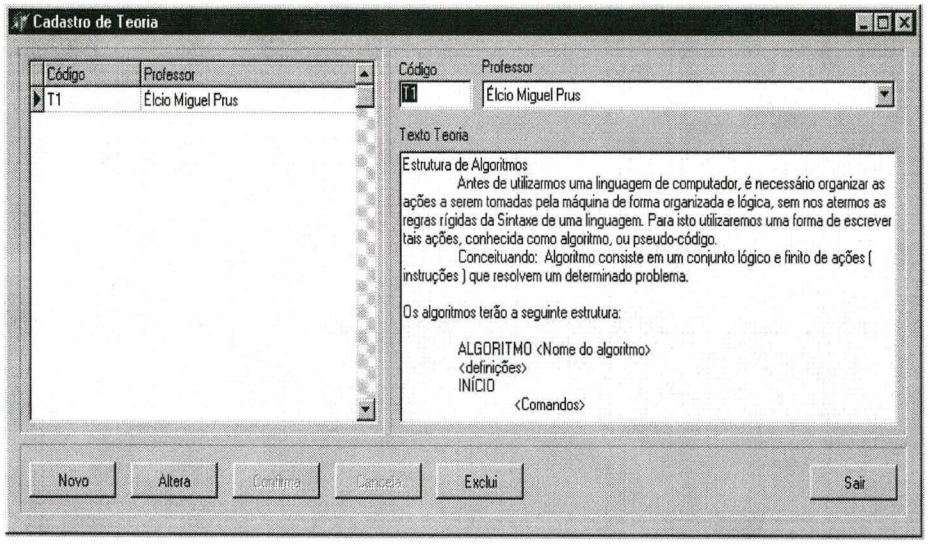



**Figura 11** – Tela de Cadastro de Teoria.

Nesta tela a inserção do código é que definirá o relacionamento apresentado na figura 10. No campo professor estarão listados todos aqueles professores cadastrados na opção de cadastro do professor. No campo texto teoria é que será inserido o texto relativo a teoria cadastrada. Se o texto da teoria a ser cadastrada estiver em um editor de textos padrão Windows®, a utilização do recurso copiar e colar, facilitará o cadastramento da mesma.

Os exercícios serão inseridos, consultados, modificados e excluídos pelo

professor através da tela de cadastro de exercícios. Cada exercício terá um código único que o identificará no sistema, estará relacionado a uma determinada teoria, possuirá um nível de dificuldade inserido pelo professor. Estará vinculado a uma disciplina e possuirá além do enunciado, que caracteriza um problema a ser resolvido através de um algoritmo estruturado, a solução proposta pelo professor. O campo Linhas Média conterá o número de linhas de código em que o exercício poderá ser resolvido. Os campos Tempo Mínimo e Tempo Máximo balizarão em quantos segundos o exercício poderá ser resolvido. A figura 12 ilustra a tela de cadastramento de exercício.

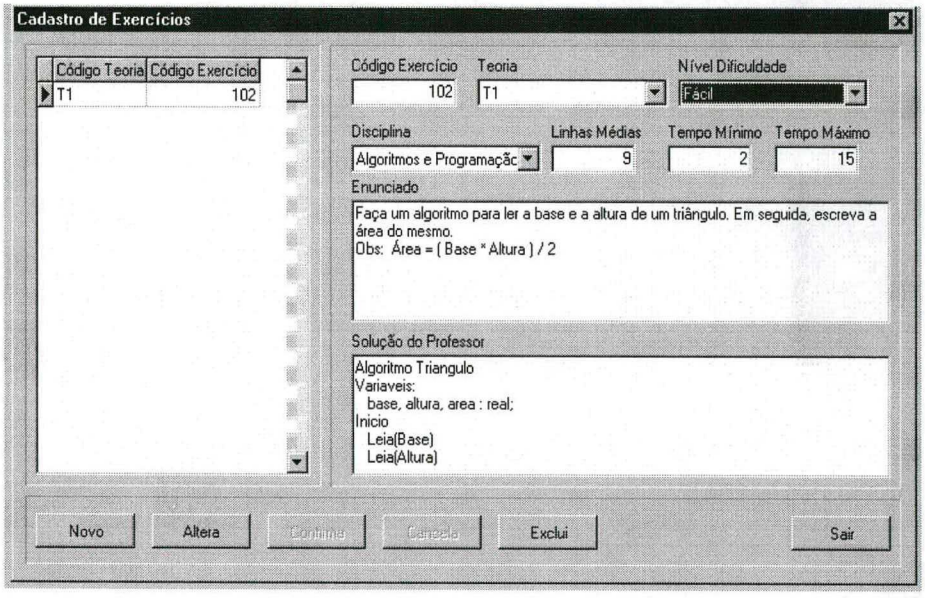



**Figura 12** – Tela de Cadastro de Exercício.

O enunciado do exercício deverá possuir um baixo grau de complexidade, pois o enunciado de um problema é um forte componente de motivação do aluno no aprendizado. Pimentel & Direne (1997) sugerem que a repetição sistemática de

enunciados completamente diferentes, porém, de graus de complexidade aproximadamente iguais podem elevar consideravelmente a autoconfiança do aprendiz, mantendo-o motivado e produtivo.

Os exercícios propostos serão classificados, pelo professor, através de sua experiência em: fácil, médio, difícil e muito difícil. Esta classificação permite ao professor quantificar e classificar sua experiência no desenvolvimento de algoritmos utilizando esta faixa de valores. O fato de não serem valores precisos, exatos, permite ao professor um melhor enquadramento dos exercícios que estarão no modelo do especialista. Este nível de dificuldade será utilizado pelo modelo pedagógico na apresentação de um exercício a ser resolvido.

### 5.3.3 O Modelo Pedagógico

O modelo pedagógico tem como principal função apresentar ao aluno o conhecimento armazenado dentro do modelo do especialista dentro de uma estratégia de aprendizagem mais adequada. Como já apresentado, o modelo proposto adotará duas estratégias de ensino aplicada a programação estruturada: a de acabamento e a de generalização. Giraffa (1998) defende a utilização de, pelo menos, de duas estratégias dentro de um STI, pois possibilita caminhos alternativos de aprendizagem.

Enquanto os exercícios a serem resolvidos forem classificados como fácil e

médio a estratégia adotada será a de acabamento. Nos exercícios classificados como difícil e muito difícil a estratégia adotada será a de generalização. Ambas serão demostradas através das telas do modelo de interface.

### 5.3.4 O Modelo de Interface

O modelo de interface é responsável pela apresentação do material instrucional ao aluno. Esta comunicação permite ao aluno ter acesso a teoria sobre estruturas básicas de programação com seus exemplos e, principalmente, aos exercícios propostos, apresentados de acordo com a estratégia de ensino escolhida pelo modelo proposto. A tela principal da interface do aluno está apresentada na figura 13.

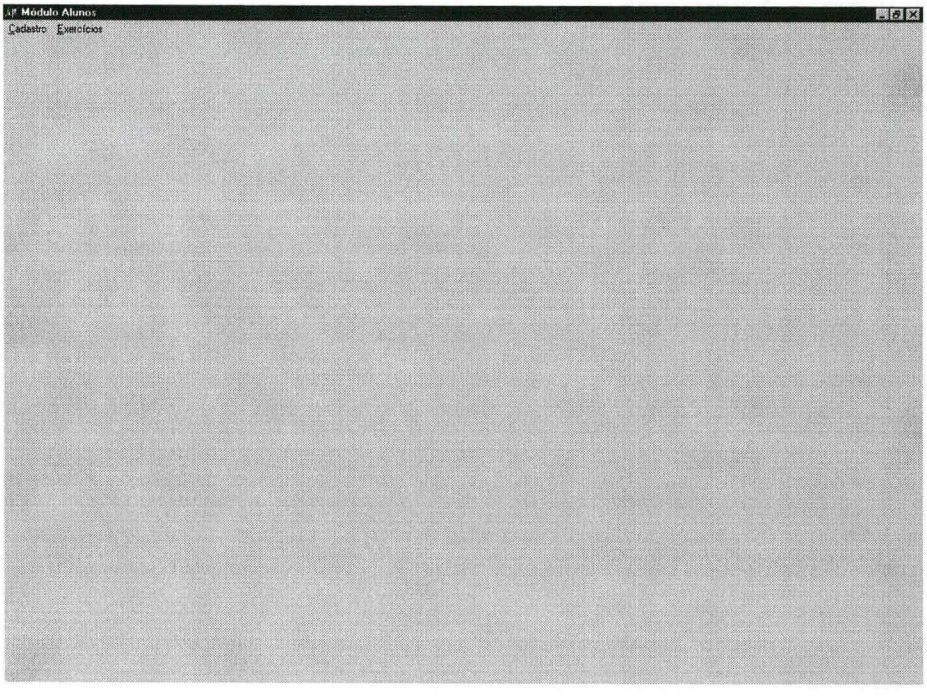



**Figura 13** – Tela do Aluno.

Nesta tela pode-se na função cadastro, colocar dados do aluno, para que o sistema possa personalizar algumas operações. Na opção exercícios tem-se o atalho para a tela de resolução dos mesmos.

No cadastro do aluno, ilustrado pela figura 14, além da matrícula e do nome, o aluno identifica a qual turma pertence, o que futuramente, possibilitará uma comparação da evolução das turmas, seu nível de experiência em relação a algoritmos estruturados. Esta informação será utilizada na formulação do primeiro exercício. A senha personalizará e somente permitirá o acesso aos exercícios do aluno identificado, reservando o acesso ao seu desempenho.

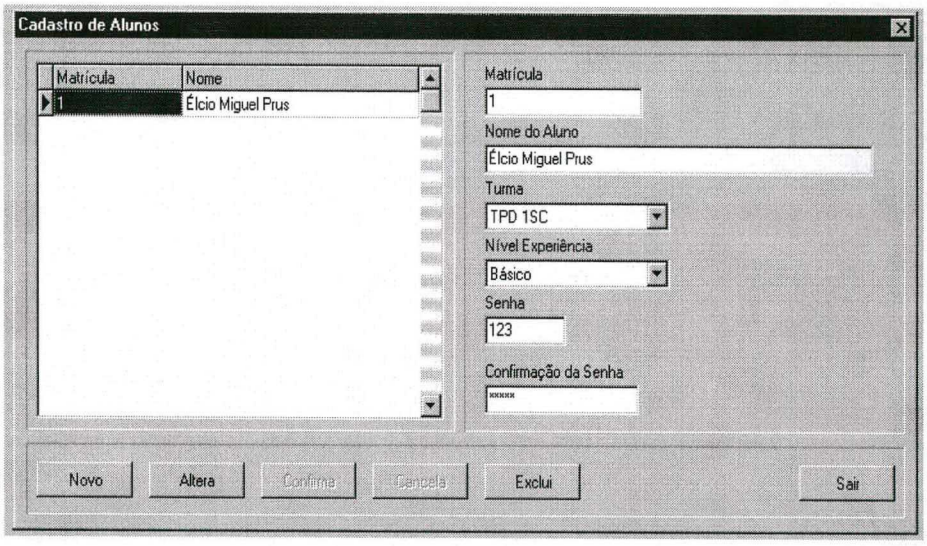



**Figura 14** – Tela de Cadastro do Aluno.

Na opção exercícios o aluno terá acesso a tela de resolução de exercícios, apresentada pela figura 15.

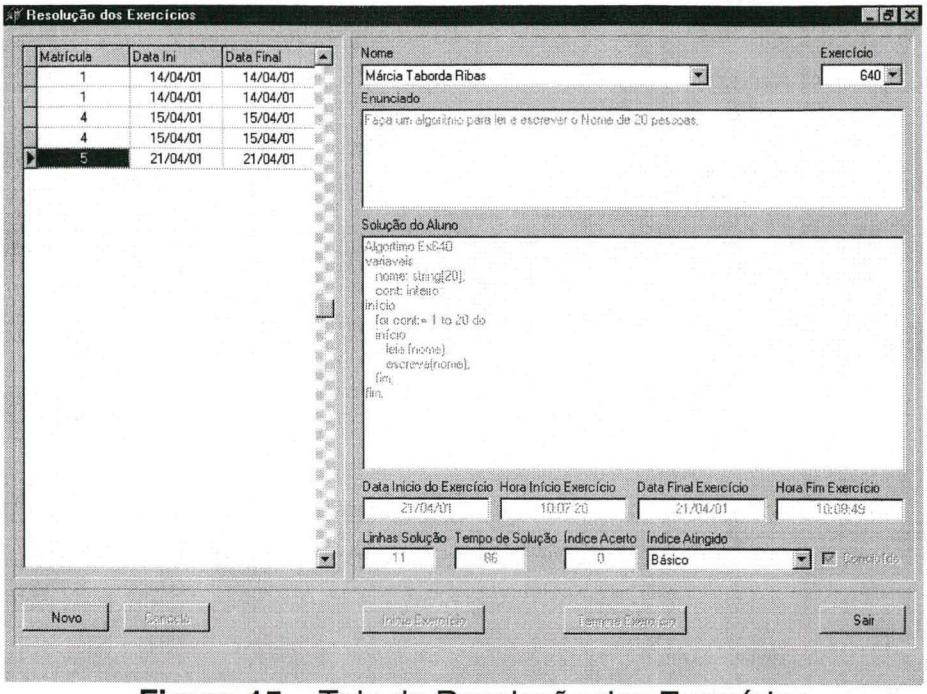



**Figura 15** – Tela de Resolução dos Exercícios.

Nesta tela o aluno, após acionar o botão Novo, seleciona o seu nome no campo Nome e em seguida o exercício a ser confeccionado, no campo Exercício. O enunciado corresponde a este exercício será apresentado no campo Enunciado. Depois da leitura do enunciado o aluno aciona o botão Inicia Exercício e os campos Data de Início do Exercício e Hora Início Exercício receberão como conteúdo a data e hora do computador, gravando-os nos campos correspondentes. Estas referências servirão como comparação para a correção.

Para resolver o problema citado no enunciado o aluno deve acionar o campo Solução, clicando com o mouse sobre o referido campo, e então, começar a resolver o que foi pedido.

Durante a solução o campo Tempo de Solução fica cronometrando o número de segundos que o aluno está utilizando para a resolução do enunciado. Quando o aluno concluir a solução deve acionar o botão Termina Exercício, quando, então o relógio que estava cronometrando o tempo pára a contagem, e os campos Data Final Exercício e Hora Fim Exercício recebem como conteúdo a data e hora atuais do computador. Estes dados ficam registrados no modelo do estudante, que armazenará a evolução do aluno nos diversos exercícios.

A diferença entre as datas e horas gravadas no sistema geram a metade do índice do acerto. O restante é calculado com a contagem do número de linhas produzidas pelo aluno na solução apresentada. Esta informação está no campo Linhas Solução. A combinação destes dois valores geram, definitivamente, o índice de acerto, que é gravado no campo Índice Acerto.

Quando finalizar um exercício o modelo proposto procederá a correção do mesmo, aplicando as regras de produção armazenadas dentro o modelo do especialista, contando o número de linhas de código produzida e cronometrando o tempo de resolução do mesmo. Estes dados são gravados indicando, data e hora do início e término de cada exercício realizado, com o número de linhas da solução apresentada. O tempo da solução é um relógio que cronometra o número de segundos que o aluno necessitou para confeccionar a solução. O número de linhas conta quantas linhas de código

foram produzidas na solução apresentada pelo aluno.

Com base nestes dois aspectos: tempo de desenvolvimento do exercício e números de linhas de código produzida, o sistema procede a correção, transformando estas duas variáveis em um índice de aproveitamento, que indicará o grau de acerto do exercício e, de acordo com a tabela 1, levará o aprendiz a um outro nível, o manterá dentro do mesmo ou ainda, fará com que retorne a teoria para uma revisão. Quanto maior for o valor deste índice mais correto estará o exercício desenvolvido.

Os valores contidos na tabela 1 estão inseridos dentro do modelo do especialista através de regras de produção. O índice de aproveitamento obtido denota o quanto o aluno acertou. Observa-se que o modelo proposto não determina um índice preciso de correção e sim um valor que representa uma faixa onde a solução proposta pelo aluno foi enquadrada de acordo com a solução apresentada pelo professor. O sistema não identifica na solução construída pelo aluno as diferenças encontradas em relação a solução proposta pelo professor.

| Índice de Aproveitamento | Índice de Acertos no exercício atual | Nível de dificuldade do próximo exercício |
|---|---|---|
| 0 | menor que 40 % | retorno a teoria |
| 1 | entre 41 e 50 % | um nível abaixo do anterior |
| 2 | entre 51 e 75 % | mesmo nível do anterior |
| 3 | entre 76 e 90 % | um nível acima do anterior |
| 4 | entre 91 e 100% | dois níveis acima do anterior |

**Tabela 1** – Regras para escolha do próximo exercício

Simultaneamente ao acionar o botão Termina Exercício, o sistema além de encerrar a contagem do tempo, abre uma nova janela com as informações relativas a teoria a qual o exercício proposto está vinculado e apresenta a solução proposta pelo professor para que o aluno possa comparar. Esta tela é apresentada na figura 16.

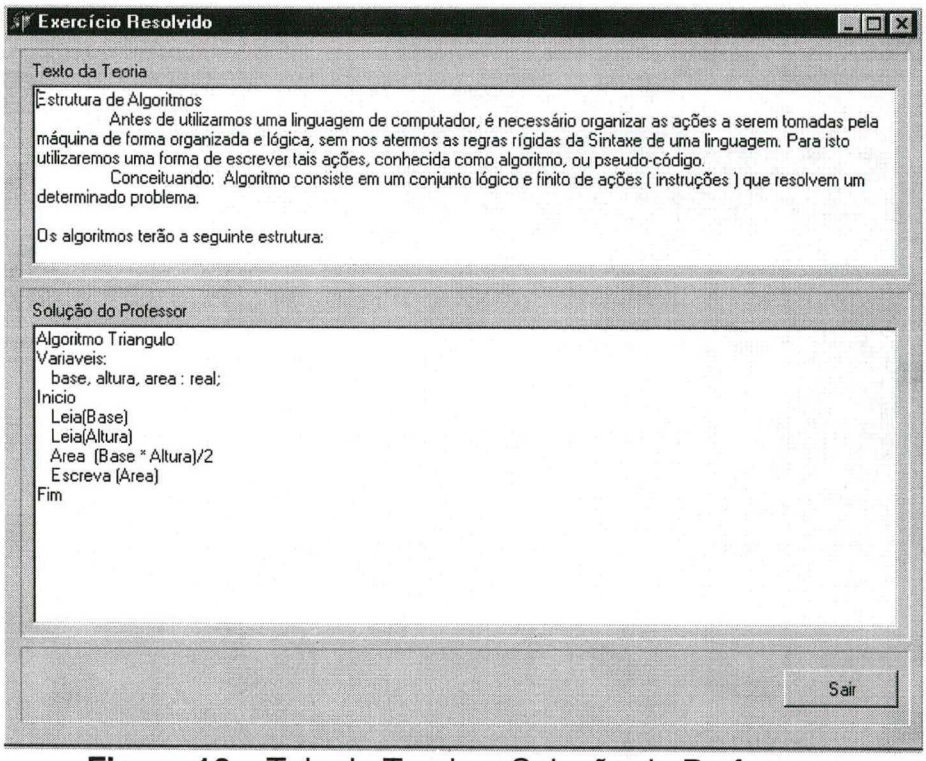



**Figura 16** – Tela de Teoria e Solução do Professor.

A apresentação desta tela tem como principal objetivo reforçar a teoria vinculada ao exercício e proporcionar ao aluno uma visualização da solução apresentada pelo professor, já que o sistema apenas indica o índice de aproveitamento obtido e não os pontos discrepantes no código entre a solução

realizada pelo aluno e a sugerida pelo professor.

### 5.3.5 O Modelo do Estudante

O modelo do estudante é responsável pelo controle da evolução dos exercícios, observando o desempenho obtido nos anteriores. Este módulo controlará o aluno desde o ponto inicial, que é um exercício classificado como fácil até o momento em que o aluno finalize o aplicativo. Em todos os exercícios propostos o sistema obedecerá as regras da tabela 1 para escolha do próximo exercício:

Para melhor ilustrar a atuação deste modelo dentro do sistema será descrita quatro situações possíveis de ocorrer que envolvem quatro, diferentes, alunos, identificados como A, B, C e D. Estes alunos iniciam a utilização do sistema, após terem um determinado conteúdo em sala de aula.

O aluno A, faz o primeiro exercício classificado como fácil, e atinge o nível de aproveitamento 3, que representa um acerto entre 75 e 90 % do exercício proposto. O aluno que estava no primeiro exercício receberá um segundo, classificado como médio. Realizado este novo exercício no qual atinge o nível de aproveitamento 3. Recebe um novo exercício, agora, classificado como difícil e realiza-o com nível de aproveitamento 3. Recebe outro exercício classificado como muito difícil. Feito este último, no qual também, atinge nível de aproveitamento 3, o modelo proposto, através, da interface com o aluno,

informa o mesmo que o nível de aprendizagem e conhecimentos demonstrados, estão acima das expectativas e após registrar estas informações no modelo do estudante, informa que o aprendiz está apto neste assunto. Graficamente, o aluno A estaria representado na figura 17.

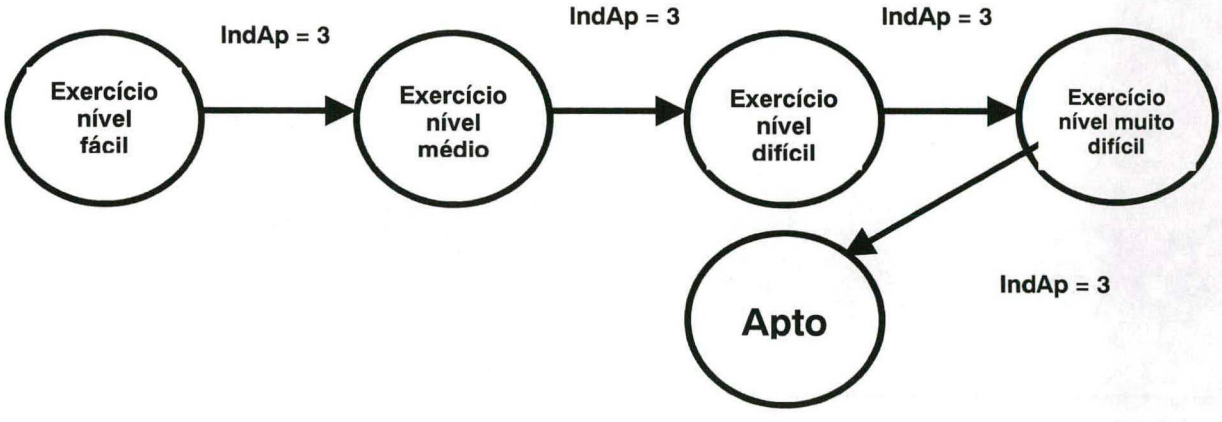



**Figura 17** – Representação gráfica da situação do aluno A.

O aluno B parte do mesmo ponto que o aluno A. Mas apresenta dificuldades na elaboração da solução, e o nível de aproveitamento identificado pelo modelo proposto foi o nível 2, ou seja, entre 50 e 75 % de acerto. O próximo exercício proposto será do mesmo nível do anterior, ou seja, fácil. Esta situação persiste por 3 (três) exercícios, e o sistema remete o aluno ao assunto relacionado na teoria, para reforço de aprendizagem e reiniciará os exercícios, registrando no modelo do estudante este acontecimento. Graficamente, o aluno B, estaria representado na figura 18.

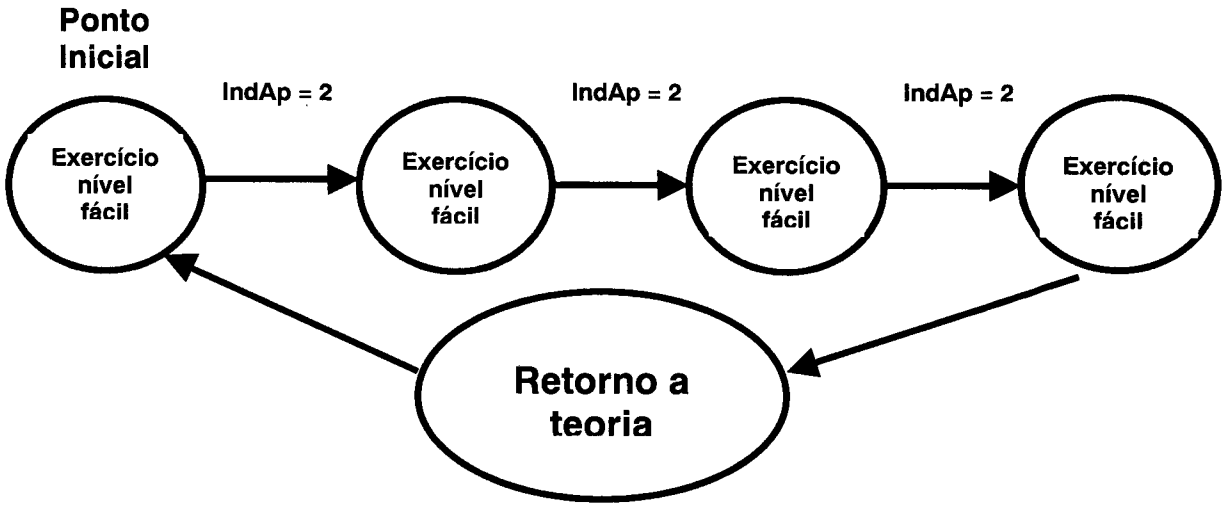



**Figura 18** – Representação gráfica da situação do aluno B.

O aluno C parte do mesmo ponto que o aluno A e B. Porém ao realizar o primeiro exercício proposto, classificado como fácil, apresenta dificuldades e obtém o nível de aproveitamento 2. O segundo exercício proposto será de mesma classificação, fácil, e neste o aluno consegue obter nível de aproveitamento 3. Recebe, então um terceiro exercício, classificado como médio, que após ser concluído, obtém nível de aproveitamento 1. O sistema identifica que o aluno não reúne condições de receber novos exercícios, nem do mesmo nível e nem de níveis mais elevados e propõe um retorno a teoria e um recomeço da elaboração dos exercícios. Graficamente, o aluno C, estaria representado na figura 19.

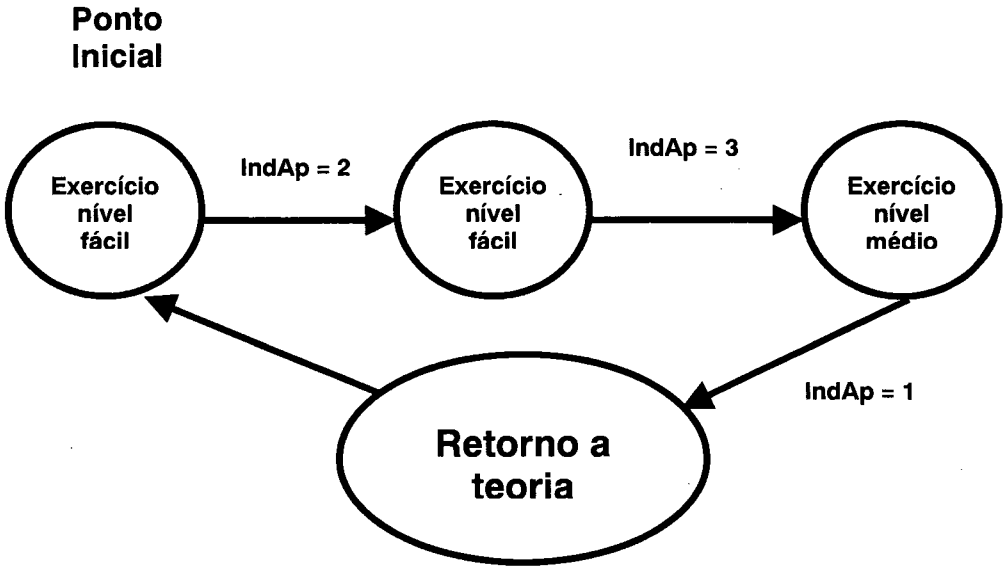



**Figura 19** – Representação gráfica da situação do aluno C.

O último aluno, D, que realizou como ponto de partida o primeiro exercício de classificação fácil e obteve como nível de aproveitamento 4, com acerto entre 90 e 100 %, recebe um segundo exercício de nível difícil, ou seja, dois níveis acima em decorrência do seu nível de aproveitamento identificado. Resolvido este segundo exercício, no qual obteve nível de aproveitamento 3, recebe outro com classificação muito difícil. Na realização deste obteve, novamente, o nível de aproveitamento 3 o que possibilita ao sistema identificar que este aluno está apto neste assunto. A representação gráfica deste aluno pode ser observada na figura 20.

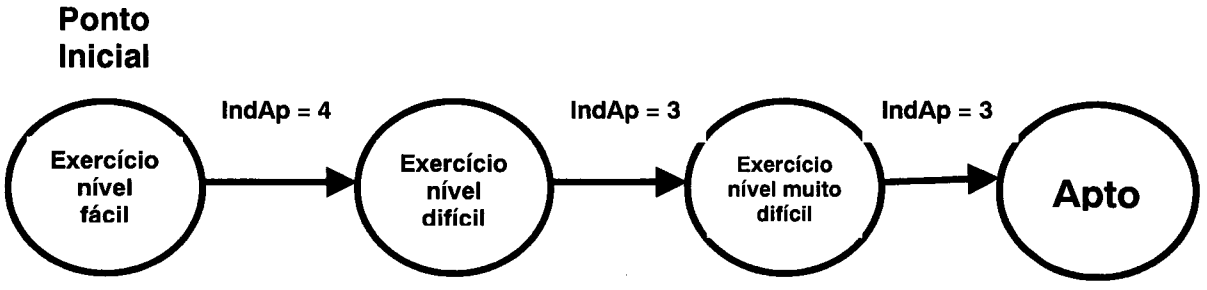



**Figura 20** – Representação gráfica da situação do aluno D.

Como o observado pela descrição das situações anteriores, o modelo proposto parte de um objetivo que é um determinado assunto dentro do domínio que representa, ou seja as estruturas básicas de programação, e possibilita ao estudante a identificação se está apto ou não em determinada estrutura, através da resolução de exercícios e da comparação destes com a solução apresentada pelo professor.

## 5.4 Considerações Finais

O modelo proposto de um STI aplicado ao ensino de programação estruturada foi modelado com base em exemplos de STI existentes e, principalmente, com base na complementação de conteúdo que a disciplina de Algoritmos e Programação I, descrita nos objetivos desta pesquisa, necessita.

A modelagem utilizou ferramentas específicas aplicadas ao processo de análise e implementação de sistemas estruturados, descritos por Yourdon

(1999), no entanto, as ferramentas de modelagem utilizadas, como o DER, não retiram a proximidade existente com o modelo proposto e os recursos de IA necessários para uma futura implementação completa do mesmo.

O modelo atende o conteúdo programático da disciplina de Algoritmos e Programação I (anexo 1). Como já citado esta disciplina é introdutória no currículo de programação que o curso de Tecnologia em Processamento de Dados desenvolve. Ela é base e pré-requisito as demais disciplinas relacionadas ao assunto de programação existentes no curso. O modelo proposto vem de encontro com a necessidade de reforçar os conteúdos abordados e efetivar uma melhor aprendizagem, considerando os conhecimentos anteriores de cada aluno.

Para Direne & Pimentel (1998) os avanços recentes na generalização de estratégias de ensino, sistemas de autoria e de shells tutores contém informações valiosas sobre arquiteturas internas que podem beneficiar muito a construção de sistemas de ensino de programação, especialmente quando o projeto de tais sistemas inclui aspectos de aprendizagem à distância.

# 6 CONCLUSÕES E RECOMENDAÇÕES

O objetivo deste capítulo é avaliar o modelo proposto e o protótipo desenvolvido de forma qualitativa, apresentando as conclusões obtidas da pesquisa realizada e propor recomendações para futuros trabalhos de continuidade da pesquisa e de implementação do STI.

## 6.1 Avaliação do Modelo Proposto

Para o desenvolvimento deste tópico, o protótipo foi apresentado a cinco alunos, todos do primeiro semestre de TPD e ao professor que trabalha com esta disciplina. A eles foi apresentado a idéia do modelo, o protótipo e então, logo após, uma rápida explicação de como operá-lo, foi dado cinqüenta minutos, o que eqüivale a uma hora-aula adotada nas Faculdades Santa Cruz, para que os mesmos pudessem utilizá-lo.

Decorrido o tempo, inicialmente sugerido, cada um teve um espaço de dez minutos para que, sem formalidades, apresentassem as impressões sobre o que avaliou. Os resultados foram anotados e separados em dois grupos: os aspectos positivos e negativos obtidos da utilização do protótipo, apresentados a seguir, considerando o professor e os alunos.

### 6.1.1 Aspectos Positivos

Os seguintes aspectos positivos foram identificados, pelo professor, no

protótipo:

1. A idéia é muito boa e adequada a disciplina, pois não existe dificuldade maior do que fazer um exercício de programação e não ter como ficar sabendo se está certo ou não;

2. É muito fácil de utilizar, o aplicativo que cabe em um disquete de 3 ½ " podendo ser instalado em qualquer computador com plataforma Pentium;

3. A interface gráfica é bem interativa e não apresentou erros durante os testes;

4. É muito interessante a forma de escolha do próximo exercício, considerando o que foi feito no anterior;

5. A consideração do programa em relação ao nível de experiência anterior transmite confiança, sabendo que os exercícios serão de acordo com o grau de conhecimento;

6. O protótipo permite a inclusão de diversas definições que foram e que serão utilizadas na disciplina, inclusive a mais apropriada;

7. Os exercícios são fáceis de serem cadastrados, tal como qualquer outra operação do módulo do professor.

Os seguintes aspectos positivos foram identificados, pelos alunos, no protótipo:

1. A idéia trouxe uma situação muito positiva em relação aos exercícios que o professor passa para resolver em casa;

2. O protótipo é de grande valia pois pode, em alguns caso, fazer o papel do professor;

3. A interface é gráfica e bem comunicativa;

4. Após a resolução do exercício além de ter-se o índice de acertos, tem-se a solução indicada pelo professor, minimizando o tempo de cópia e auxiliando a comparação do que foi feito;

5. O objetivo proposto pela idéia é inteiramente abrangida no protótipo.

### 6.1.2 Aspectos Negativos

Os seguintes aspectos negativos foram identificados, pelo professor, no protótipo:

1. O sistema poderia considerar que um enunciado (exercício) pode possuir mais que uma solução e dar espaço para este tipo de possibilidade;

2. As telas são cinza e não apresentam indicação de como preencher determinados campos;

3. No módulo do aluno a opção de cadastro não está suficientemente clara, o que pode ocasionar dúvidas, tanto no preenchimento como na utilização;

4. A indicação do tempo de solução no campo indicado para isto pode causar um certo desconforto ao aluno.

Os seguintes aspectos negativos foram identificados, pelos alunos, no protótipo:

1. Não há nenhuma animação, como som ou imagem, o que tornou o teste um pouco monótono;

2. O campo Tempo de Solução não deveria ficar marcando o tempo de desenvolvimento, pois de uma forma ou outra pressiona a terminar o exercício;

3. A forma de correção utiliza poucos critérios e pode não indicar, com certeza, se a solução apresentada está, realmente, certa ou não;
4. O programa é pouco interativo com o aluno, se preocupando, apenas, com a questão do ensino de programação estruturada.

## 6.2 Conclusões

A mente humana não somente possui a capacidade de adquirir conhecimento, como também é capaz de manipular um nível de entendimento muitas vezes baseando-se apenas no próprio conhecimento. Apesar disto, muitos dos trabalhos na área de IA têm buscado modelar a mente humana através de uma heurística baseada em regras e não têm obtido muito êxito.

Se os computadores pudessem de alguma forma ser dotados de algo como a compreensão, introspecção e razão humana, o resultado provavelmente não seria um modelo preciso do raciocínio humano, mas certamente poderia se relacionar e interagir com humanos de uma forma muito parecida a um professor humano. O campo da pedagogia baseada em IA busca exatamente isto.

Tomando por base o objetivo geral proposto de projetar e desenvolver um modelo de STI abrangendo a matéria de Algoritmos e Programação para ser aplicado à disciplina de Algoritmos e Programação I, no curso de TPD, das Faculdades Santa Cruz de Curitiba, conclui-se que:

1. O protótipo proporciona automação na escolha do próximo exercício de algoritmo, considerando o estágio apresentado na resolução do anterior, simulando, com restrições, a escolha que seria feita por um professor;

2. A forma de apresentação da teoria, permite ao aluno, desenvolver diversas visões do mesmo conceito ou tópico de algoritmos estruturados;

3. O número de exercícios existentes no protótipo servem de auxílio e estímulo, tanto ao professor como ao aluno, pois não possuem qualquer restrição em relação a quantidade e nível de dificuldade;

4. Com base na avaliação realizada, foi possível identificar a satisfação dos alunos em operar um programa que, de alguma forma, auxilia no aprendizado da disciplina;

5. Apresenta uma interface gráfica e interativa que proporciona boa comunicação com o aluno e professor;

6. Apresenta facilidades operacionais, o que torna a concentração do usuário exclusiva para a resolução dos exercícios;

7. Há necessidade de ampliar o número de fatores considerados para a correção, minimizando a possibilidade de erro durante a mesma;

8. A forma de apresentação dos exercícios transmite ao aluno confiança, pois o mesmo sabe o próximo exercício será um de nível adequado aos conhecimentos demostrados no anterior.

### 6.3 Recomendações

As recomendações sugeridas em relação a pesquisa e ao protótipo

desenvolvido são:

1. Ampliar a análise dos currículos de cursos superiores de informática, para que a disciplina que desenvolve a habilidade de programação seja profundamente dissecada, e seja traçado um panorama, detalhado, do profissional a ser formado e da importância desta disciplina em seu perfil. Esta análise poderia ser regionalizada considerando uma área de atuação, por exemplo uma cidade, um estado ou uma macro-região;

2. Considerando, ainda, a recomendação anterior, a análise, também, deve considerar as bibliografias utilizadas na disciplina, pois ela serão a base de alimentação dos conhecimentos no modelo do especialista;

3. Ampliar o número de alunos que realizaram a avaliação para obter, de forma quantitativa, os aspectos que o protótipo deva ser melhorado;

4. Reestruturar o *design* das telas, tornando-as mais agradáveis e utilizar cores e recursos multimídias para interagir melhor com o usuário;

5. Desenvolver um *help* (ajuda) interativo indicando a forma de preenchimento dos campos e englobando as dúvidas mais comuns que possam acontecer;

6. Reavaliar a linguagem utilizada para o desenvolvimento do sistema. No caso a linguagem Delphi® versão 5.0 foi aplicada, porém, versões mais atualizadas e direcionadas para a área de IA, poderiam, com menos linhas de código, implementar mais funções, dando ao sistema mais inteligência;

7. Permitir ao professor um controle de evolução do desempenho de seus alunos, por turma.

# 7 REFERÊNCIAS BIBLIOGRÁFICAS


ALVES, Maria Bernardete Martins, SIMÕES, Priscyla Waleska T. de Azevedo. Tutoriais inteligentes, 1999 .Disponível na Internet: http://wwwedit.inf.ufsc.br:1998/alunos99/trabfinal/tutoriais.htm. (acessado em 04.07.2000)

BARRETO, Jorge Muniz. Inteligência artificial no limiar do século XXI. Florianópolis: PPP Edições, 1998.

BARROS, Célia Silva Guimarães. Psicologia e construtivismo. São Paulo: Ática, 1996.

BARROS, Célia Silva Guimarães. Pontos da psicologia escolar. São Paulo: Ática, 1998.

BITTENCOURT, Guilherme. Inteligência artificial: ferramentas e teorias. Florianópolis: Editora da UFSC, 1998.

BROWN, M. Exploring Algorithms Using Balsa-II. USA: IEEE Computer - Number 21, pág 14-36, 1988.

CHAIBEN, Hamilton. Inteligência artificial na educação, 1998. Disponível na Internet. http://www.cce.ufpr.br/~hamilton/iaed/iaed.htm. (acessado em 03.12.1999)

CHANG,K.E.; CHIAO,B.C.;HSIAO,R.S. A programming Learning System for Beginners - A Completion Strategy Approach. International Conference on Intelligent Tutoring Systems – ITS 96, Volume 3, 1996, Berlim: Springer Verlag, 1996.

CORMEN, Thomas H.; LEISERSON, Charles E.; RIVEST, Ronald L. Introduction to Algorithms. EUA: McGraw-Hill Book Company, 1990.

CÓRIA-SABINI, Maria Aparecida. Psicologia aplicada à educação. São Paulo: EPU, 1986.

COSTA, Rosa Maria E.M., WERNECK, V.M.B. Tutores Inteligentes. Rio de Janeiro: COPPE /UFRJ, 1996.

DAVIS, Cláudia; OLIVEIRA, Zilma de Moraes Ramos. Psicologia na educação. São Paulo: Cortez, 1994.

DILLENBOURG, Pierre. The role of artificial intelligence techniques in training software, 1994. Disponível na Internet. http://tecfa.unige.ch/tecfa/general/tecfa-people/dillenbourg.html (acessado em 10/01/2001)

DIRENE, Alexandre Ibrahim; PIMENTEL, Andrey Ricardo. Medidas cognitivas no ensino de programação de computadores com sistemas tutores inteligentes. In: IX Simpósio Brasileiro de Informática na Educação (SBIE), 9., Anais..., Fortaleza, 1998.

EBERSPÄCHER, Henri Frederico. Proposta, Projeto e Desenvolvimento de um Sistema de Autoria para Construção de Tutores Inteligentes Hipermídia. Curitiba, abril de 1998. (Dissertação de Mestrado apresentado ao Curso de Pós-Graduação em Engenharia Elétrica e Informática Industrial do Centro Federal de Educação Tecnológica do Paraná).

FALCÃO, Gérson Marinho. Psicologia da aprendigazem. São Paulo: Ática, 1996.

FARRER, Harry et alli. Pascal estruturado. 3ª edição. Rio de Janeiro: LTC, 1999.

FIALHO, Francisco Antônio Pereira. Uma introdução à engenharia do conhecimento, 1998. Disponível na Internet: http://eps.ufsc.br/disciplinas/fialho/ergcog/index.html. (acessado em 06/07/2000)

GIRAFFA, Lucia Maria Martins. Fundamentos de sistemas tutores inteligentes, 1996. Disponível na Internet: http://www.inf.pucrs.br/~giraffa/ia/textosia.html. (acessado em 03.07.2000)

GIRAFFA, Lucia Maria Martins. Uma arquitetura de tutor utilizando estados mentais. Porto Alegre, 1998. (Tese de Doutorado apresentado ao Curso de Pós-Graduação em Ciência da Computação da Universidade Federal do Rio Grande do Sul).

GUIMARÃES, Ângelo de Moura; LAGES, Newton Alberto de Castilho. Algoritmos e estruturas de dados. Rio de Janeiro: LTC, 1985.

JONASSEN, D.H.,WANG, S., The Physics Tutor: Integrating Hypertext and Expert Systems. Journal of Educational Technology Systems, Vol. 22, p. 19-28, 1993.

KAPLAN, Randy e ROCK, Denny. New directions for Intelligent Tutoring Systems. AI Expert, Fev. 1995, p. 31-40.

KEARSLEY, Greg. Artificial Intelligence and Instruction, applications and methods. Addison Wesley, 1987.

LIBERMAN, H. Seeing What Your Programs are Doing. USA: International Journal of Man-Machine Studies, pág 311-331, 1984.

MANBER, Udi. Introduction to algorithms: a creative approach. EUA: Addison-Wesley Company Inc, 1989.

MATUI, Jiron. Construtivismo: teoria construtivista sócio-histórica aplicada ao ensino. São Paulo: Moderna, 1995.

MINISTÉRIO DA EDUCAÇÃO E CULTURA. Secretaria de Educação Superior. Diretrizes curriculares de cursos da área de computação e informática. Brasília: MEC, 1999.

MIZUKAMI, Maria da Graça Nicoletti. Ensino: as abordagens do processo. São Paulo: EPU, 1986.

MOREIRA, Marco Antonio. Teorias de aprendizagem. São Paulo: EPU, 1999.

NISKIER, Arnaldo. LDB: a nova lei da educação: tudo sobre a lei de diretrizes e bases da educação nacional: uma visão crítica. Rio de Janeiro: Consultor, 1996.

OLIVEIRA, Martha Kohl de. Vygotsky: aprendizado e desenvolvimento: um processo sócio-histórico. São Paulo: Scipione, 1997.

PIMENTEL, A.; DIRENE, I. Medidas Cognitivas para o Ensino de Conceitos Visuais com Sistemas Tutores Inteligentes. Anais do VIII Simpósio Brasileiro de Informática na Educação (SBIE-97): São José dos Campos, 1997.

PINTO, Wilson Silva. Introdução ao desenvolvimento de algoritmos e estrutura de dados. São Paulo: Érica, 1990.

RABUSKE, Renato Antônio. Inteligência artificial. Florianópolis: Editora da UFSC, 1995.

RICH, Elaine. Inteligência artificial. São Paulo: McGraw Hill, 1988.

RICH, Elaine; KNIGHT, Kevin. Inteligência artificial. São Paulo: Makron Books do Brasil, 1994.

SALVETTI, Dirceu Douglas; BARBOSA, Lisbete Madsen. Algoritmos. São Paulo: Makron Books, 1998.

SEBESTA, Robert W. Conceito de linguagens de programação. Tradução de José Carlos Barbosa dos Santos. 4ª Edição. Porto Alegre: Bookman, 2000.

SILVA, Valdete T. Módulo pedagógico para um ambiente hipermídia de aprendizagem. Florianopólis, 2000. (Dissertação de Mestrado apresentado ao Programa de Pós-Graduação em Engenharia da Produção da Universidade Federal de Santa Catarina).

SILVEIRA, R.A. Inteligência Artificial em Educação: um modelo de sistema tutorial inteligente para microcomputadores. Porto Alegre, 1998. (Dissertação de Mestrado apresentado ao Curso de Pós-Graduação em Ciência da Computação da Pontífice Universidade Católica do Rio Grande do Sul).

SILVEIRA,R.A. ELETROTUTOR II : um tutor na WEB. Porto Alegre: UFRGS, 1996.

SZWARCFITER, Jayme Luiz; MARKEZON, Lilian. Estruturas de dados e seus algoritmos. 2ª Edição Revista. Rio de Janeiro: LTC, 1994.

TERADA, Routo. Desenvolvimento de algoritmos e estruturas de dados. São Paulo: McGraw-Hill, Makron, 1991.

TERADA, Routo; SETZER, Valdemar W. Introdução à computação e à construção de algoritmos. São Paulo: Makron Books, 1992.

TREMBLAY, Jean-Paul; BUNT, Richard B. Ciência dos computadores: uma abordagem algorítmica. São Paulo: McGraw-Hill do Brasil, 1983.

VICCARI, Rosa Maria. Sistemas Tutores Inteligentes: abordagem tradicional x abordagem de agentes. Simpósio Brasileiro de Inteligência Artificial. Curitiba, 1996.

WARNIER, Jean Dominique. LCP: lógica de construção de programas. Rio de Janeiro: Campus, 1991.

WENGER, E., Artificial Intelligence and Tutoring Systems. USA: Morgan Kaufmann Publishers, 1987.

WINSTON, Patrick H. Inteligência artificial. São Paulo: Livros Técnicos e Científicos, 1987.

WIRTH, Niklaus. Algoritmos e estrutura de dados. Rio de Janeiro: Prentice-Hall do Brasil, 1989.

WOOLF, B. Intelligent Tutoring Systems: a survey. National Conferences on Artificial Intelligence, 1988. USA: Morgan Kaufmann Publishers, pág.1-43.

YOURDON, E. Análise Estruturada Moderna. São Paulo: Campus, 1999.

ZIVIANI, Nivio. Projeto de algoritmos: com implementações em Pascal e C. 4ª Edição. São Paulo: Pioneira, 1999.

# 8 ANEXOS

## 8.1 Anexo 1 – Conteúdo Programático da Disciplina de Algoritmos e Programação I

**FACULDADE REGIONAL SANTA CRUZ DE CURITIBA - FARESC**
*Rua Pedro Bonat, 103 - Fone/fax: (41) 248-0311 - CEP: 81110-040 - Curitiba – PR*

## *CONTEÚDO PROGRAMÁTICO*

Curso.....: ***Tecnologia em Processamento de Dados***
Disciplina.....: ***Algoritmos e Programação I***
Carga Horária.....: ***72 horas***
Departamento.....: ***Processamento de Dados***

**Objetivo(s)**
Motivar o aluno a solucionar problemas aplicando as técnicas do pensamento algorítmico, traduzindo os esparsos e complexos pensamentos em lógicos e simples, essenciais à confecção de programas computacionais. Criar o hábito da resolução de problemas através da aplicação das estruturas básicas de programação. Introduzir o aluno, na programação, através da aplicação da Linguagem PASCAL na resolução dos problemas.

**Ementa**
Desenvolver as noções básicas de lógica computacional. Funções matemáticas e lógicas. Aplicar as estruturas utilizadas em algoritmos para a resolução de problemas matemáticos, através dos conceitos do português estruturado. Estruturas de decisão, seleção e controle. Aninhamento de estruturas. Conceitos, desenvolvimento e aplicação da Linguagem PASCAL.

**Programa**
1. Introdução à Lógica Computacional: Conceitos básicos.

2. Funções Matemáticas: Tipos primitivos; Constantes; Variáveis; Expressões e Operadores matemáticos.

3. Funções Lógicas: Operadores relacionais e operadores lógicos.

4. Algoritmos: Definições e estrutura dos Algoritmos.

5. Português Estruturado: Comando de Atribuição (:=); Comando de Entrada de Dados (leia); Comandos de Saída de Dados (escreva e imprima); Estruturas ( seqüência, se-então-senão, ses encaixados, escolha-caso, enquanto-faça, para-faça, repita-Até); Aninhamento e múltiplas estruturas.

6. A Linguagem PASCAL: Conceitos básicos; Estrutura dos programas em PASCAL; O ambiente de programação; As principais bibliotecas; Estrutura dos programas em PASCAL; Comando de Atribuição (:=); Comandos de Entrada de Dados (read e readln); Comandos de Saída de Dados (write e writeln); Estruturas (Seqüência, if-then-else, if's encaixados, case-of-else, while-do, for-do, repeat-until); Aninhamento e múltiplas estruturas.

**Bibliografia**

**1. Livro(s) Texto(s):**

FARRER, Harry et alli. Pascal estruturado. 3ª edição. Rio de Janeiro: LTC, 1999.

SALVETTI, Dirceu Douglas; BARBOSA, Lisbete Madsen. Algoritmos. São Paulo: Makron Books, 1998.

TERADA, Routo; SETZER, Valdemar W. Introdução à computação e à construção de algoritmos. São Paulo: Makron Books, 1992.

TREMBLAY, Jean-Paul; BUNT, Richard B. Ciência dos computadores: uma abordagem algorítmica. São Paulo: McGraw-Hill do Brasil, 1983.

WIRTH, Niklaus. Algoritmos e estrutura de dados. Rio de Janeiro: Prentice-Hall do Brasil, 1989.

**2. Bibliografia Complementar:**

GOTTFRIED, Byron S. Programação em pascal. São Paulo: McGraw-Hill do Brasil, 1988.

INSTITUTO BRASILEIRO DE PESQUISA EM INFORMÁTICA. Técnicas de programação com pascal. Rio de Janeiro: IBPI, 1993.

KERNIGHAN, Brian W; PLAUGER, P. J. Ferramentas para a programação em pascal. Rio de Janeiro: Campus, 1988. (tradução de Fernando Cabral)

LIMA, Vera Lúcia Strube de. Linguagem pascal. 4ª edição. Rio de Janeiro: Campus, 1987.

PINTO, Wilson Silva. Introdução ao desenvolvimento de algoritmos e estrutura de dados. São Paulo: Érica, 1990.

SCHMITZ, Eber Assis; TELES, Antonio Anibal de Souza. Pascal e técnicas de programação. 3ª Edição. Rio de Janeiro: LTC, 1988.

WIRTH, Niklaus. Programação sistemática em pascal. 6ª edição. Rio de Janeiro: Campus, 1989.

ZIVIANI, Nivio. Projeto de algoritmos: com implementações em Pascal e C. 4ª Edição. São Paulo: Pioneira, 1999.